

DBMS

1. Define DBMS ?

A. DBMS is a collection of data & set of program to access store those data in an easy & efficient manner. DBMS is a software which is used to manage data base.

PBMS :- Data base Management system.

2. What are the advantages of DBMS?

A. 1. Data integrity & security

2. Data independence

3. Data administration

4. Efficient data access

5. concurrent access & crash recovery.

6. Reduce application develop time.

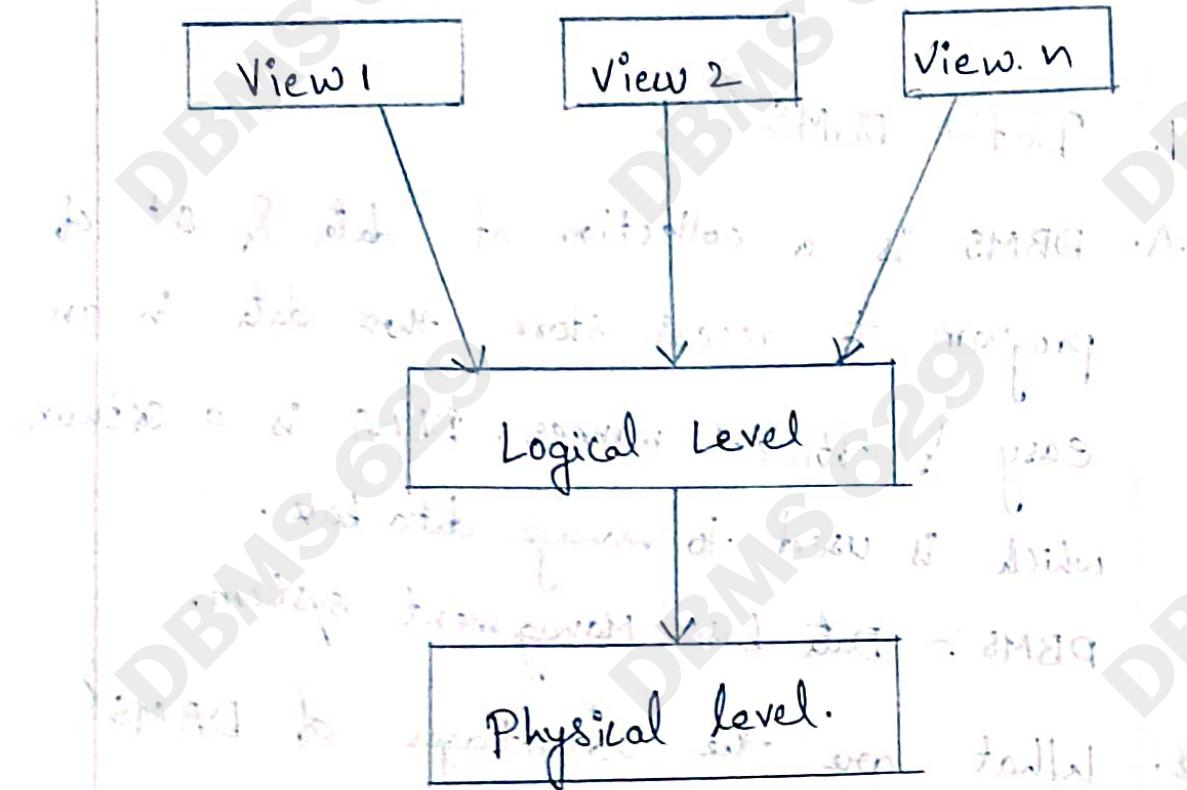
3. Explain about data base Abstraction?

A. Database systems are made up of

Complex data structures. To easy the user interaction with database, the developers hide

internal irrelevant details from users.

The process of hiding the details from user is called data abstraction.



Physical level :- This is the lowest level of data abstraction. It describes how data is actually stored in database. You can get the complex data structure details at this level.

Logical level :- This is the middle level of 3 level data abstraction architecture. It describes what data is stored in database.

View level :- This is the highest level of abstraction. This level describes the user interaction with database system.

Ex:- View level: It is just user interact with system with the help of GUI and enter the details of the screen, they are not aware of how the data is stored and what data is stored such details are hidden from them.

Logical level: Can be described as fields and attributes along with their data types, their relationship among each other can be logically implemented. The programmers generally work at this level because they are aware of such things about data base system.

Physical level: At physical level these records can be described as blocks of storage (bytes, TB, KB, MB) in memory. These details are after hidden from the programmers.

Q. Write the advantages and disadvantages of flat file Management System and Data Base Management system?

A. File management systems are providing the following advantages and disadvantages.

Advantages	Disadvantages
Simple to use.	Typically does not support multi-user access.
Less expensive.	Limited to smaller databases.
Fits the needs of many small businesses and home users.	Limited functionality.
popular FMS's packaged along with the operating system of personal computers.	Decentralization of data
Good for database solutions for hand held devices such as palm pilot.	Redundancy and integrity issues.

Data base Management systems provide the

following advantages and Disadvantages:-

Advantages	Disadvantages
Greater flexibility	Difficult to learn.
Good for large databases.	packaged separately from the operating.
Fits the needs of many medium to large-sized organizations.	Requires skilled administrators.
Storage for all relevant data.	Expensive.

What is Data Base Model?

A data model is a collection of concepts that can be used to describe the structure of the data base. Data models include a set of basic operations for specifying retrieval and update on the data base. Data models are a collection of conceptual tools for describing data, data relationships, data semantics and data constraints. There are three different groups:-

1. Object-based Logical Models.

2. Record-based Logical Models.

3. Physical Data Models.

What is object based Logical Models?

* Describe data at the conceptual and view levels.

* provide fairly flexible structuring capabilities.

* Allow one to specify data constraints explicitly.

* Over 30 such models, Including

→ Entity relationship model.

→ Object-oriented model

→ Binary model.

→ Semantic data model

→ Infological model

→ Functional data model.

Entity - Relationship Model :-

It is based on the notion of the real world entities and their relationships while formulating the real world scenario in to the data base model an entity set is created and this model is dependent on two vital things and they are:

- * Entity and their attributes.

- * Relationships among entities.

An entity has a real world property called attribute and attribute defined by a set of values called domain.

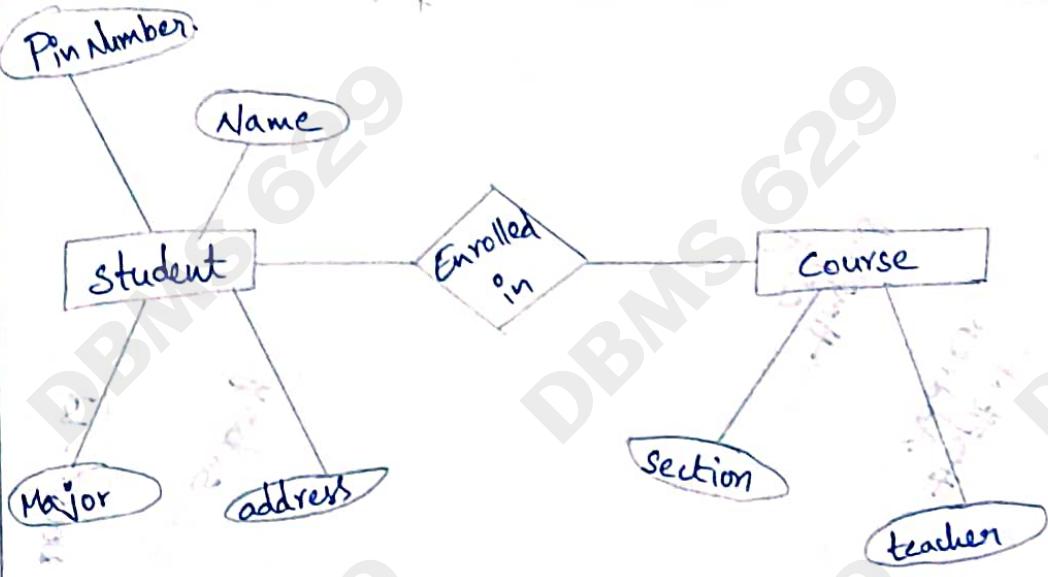
The overall logical structures of a database can be expressed graphically by an diagram.

- * Rectangles:- Represent entity sets

- * Ellipses :- Represent attributes

- * Diamonds :- Represents relationships among entity sets

- * Lines:- Link attributes to entity sets and entity sets to relationships.



Attributes:- Entities are represented by means of their properties called Attributes.

Ex:- student,

. name, class, Age, Ph.no.

→ Attributes.

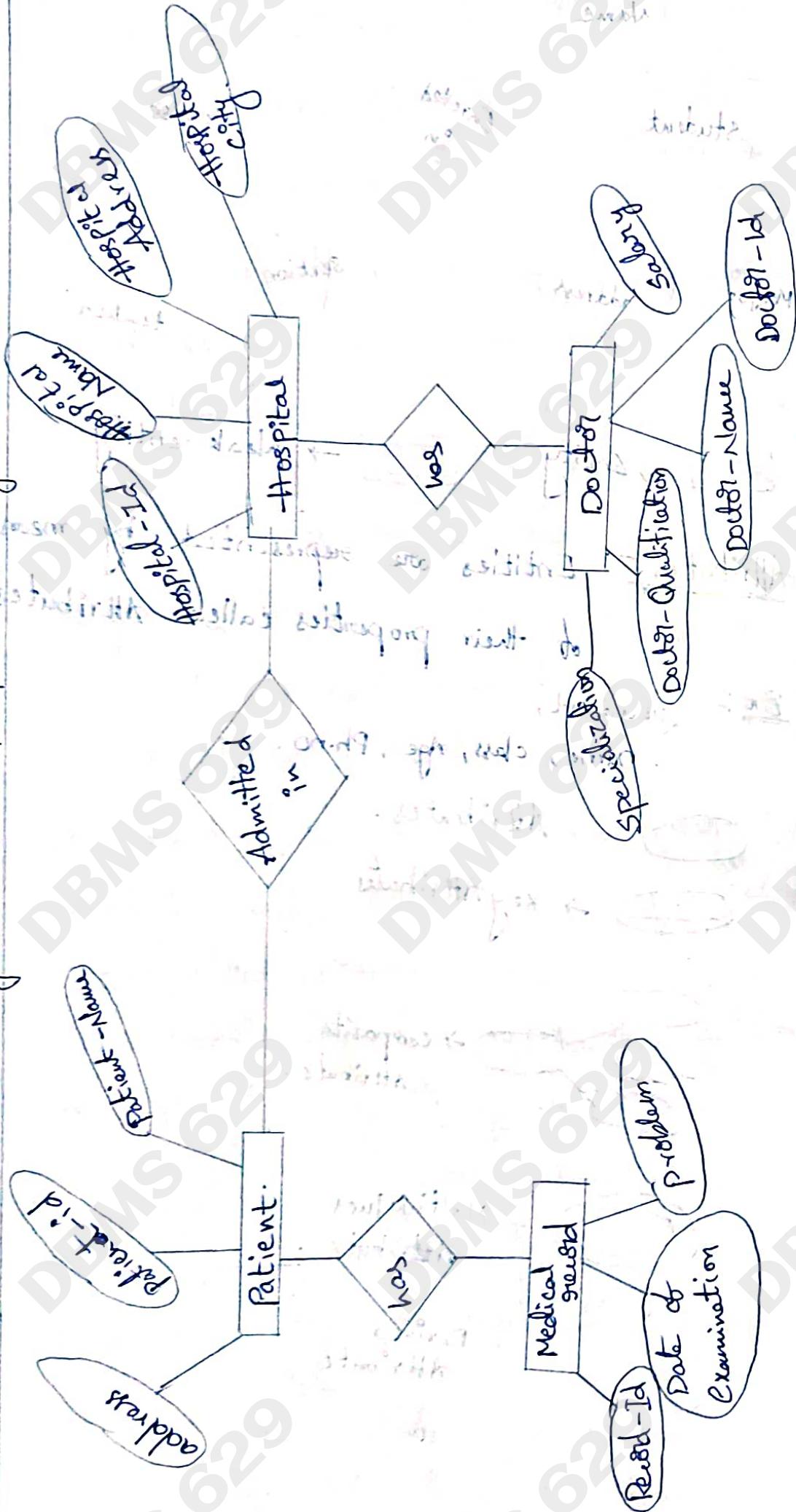
→ key Attributes

→ composite attribute

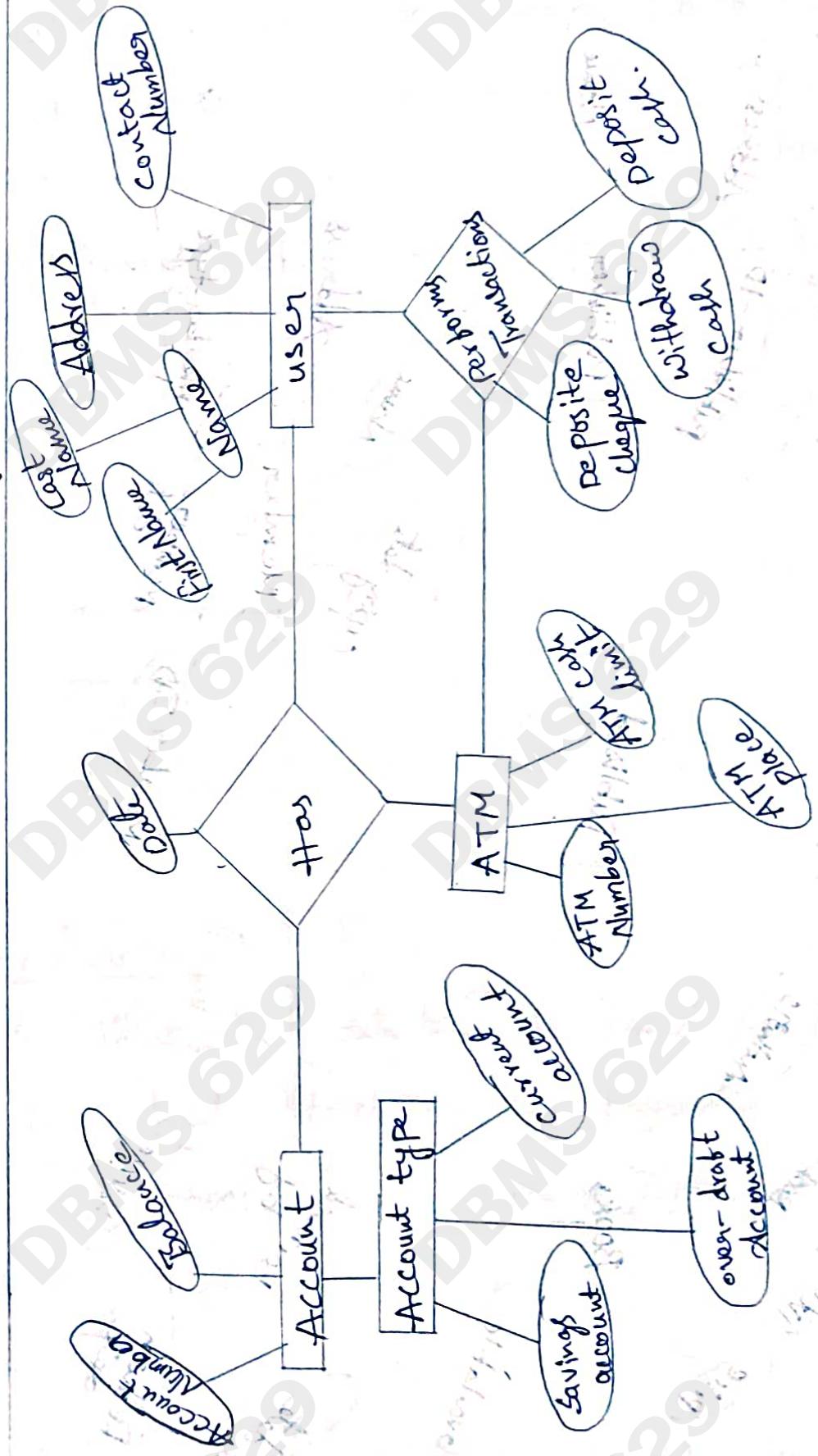
→ Multivalued Attribute.

→ Derived Attribute

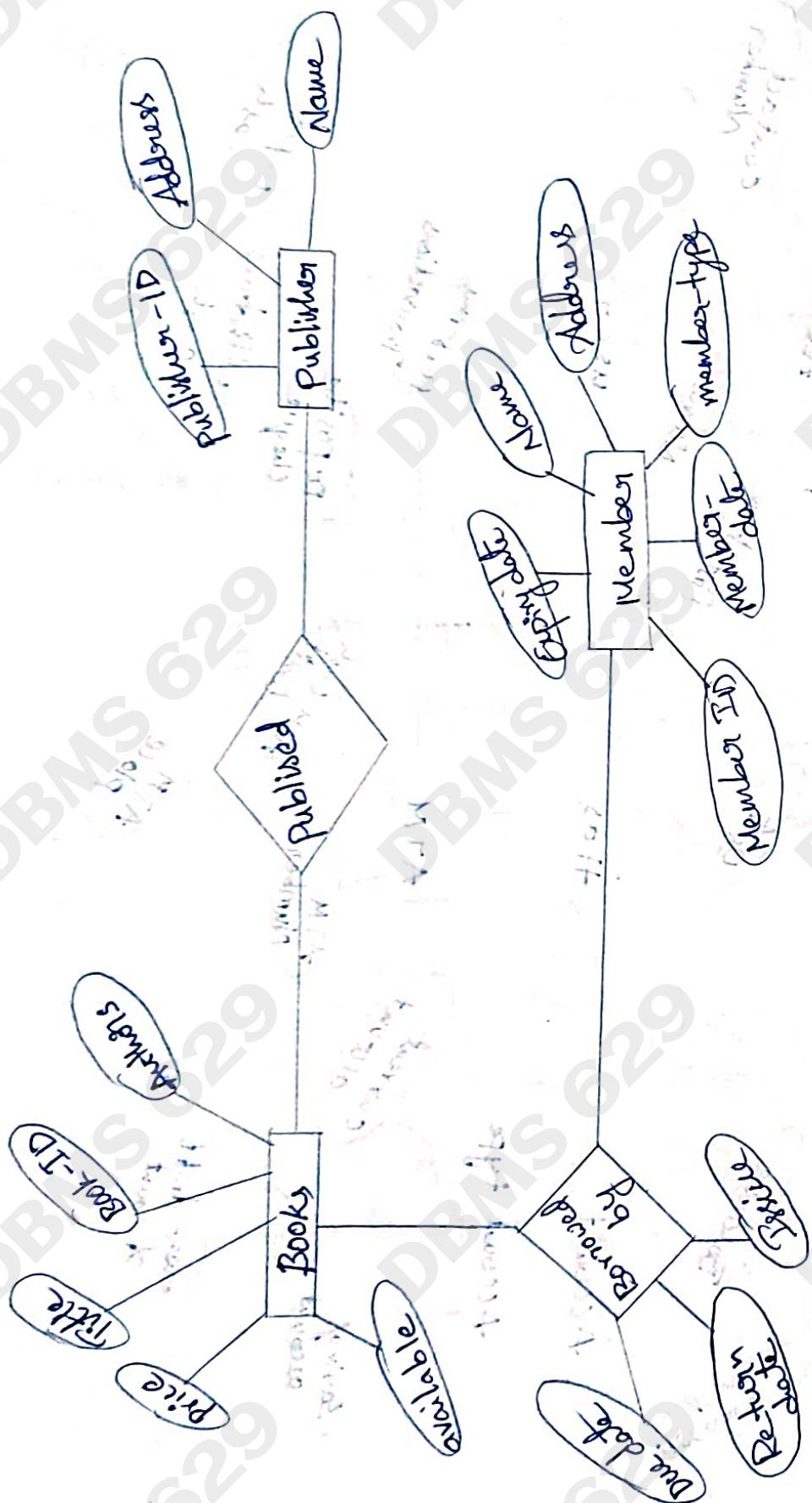
E-R Diagram of Hospital Management



E-R Diagram of Banking system.



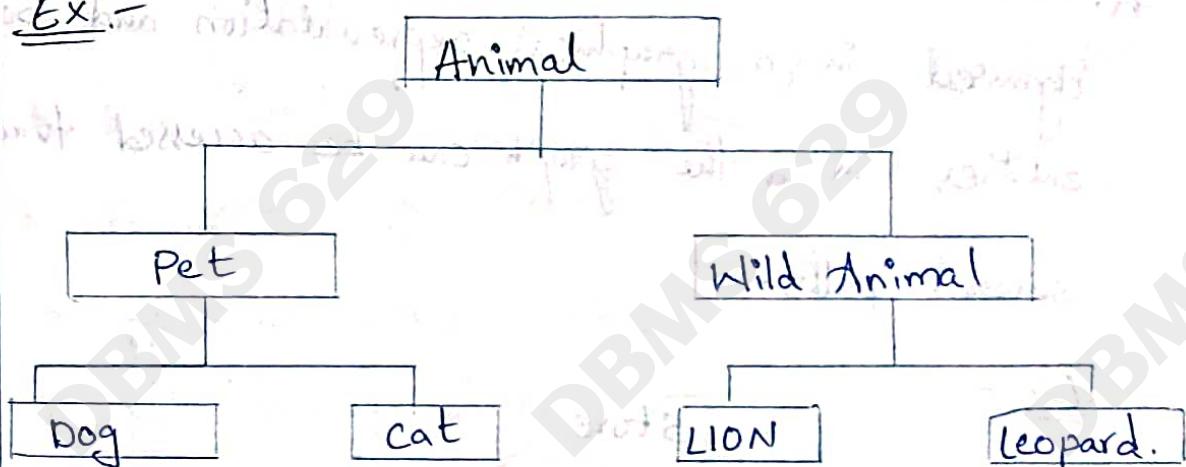
E-R Diagram of Library Management System.



What is Hierarchical Data Model?

- A. It has one parent entity with several children entity but at with several children entity but at the top we should have only one entity called root. For example department is the parent entity called root and it has several children entities like students, professors etc.

Ex:-



Advantages:-

- * Since the database is based on the hierarchical structure, the relationship between the various layers is logically simple.
- * Very efficient for one to many relationships.
- * As there is a link between parent and child, the model promotes data integrity.

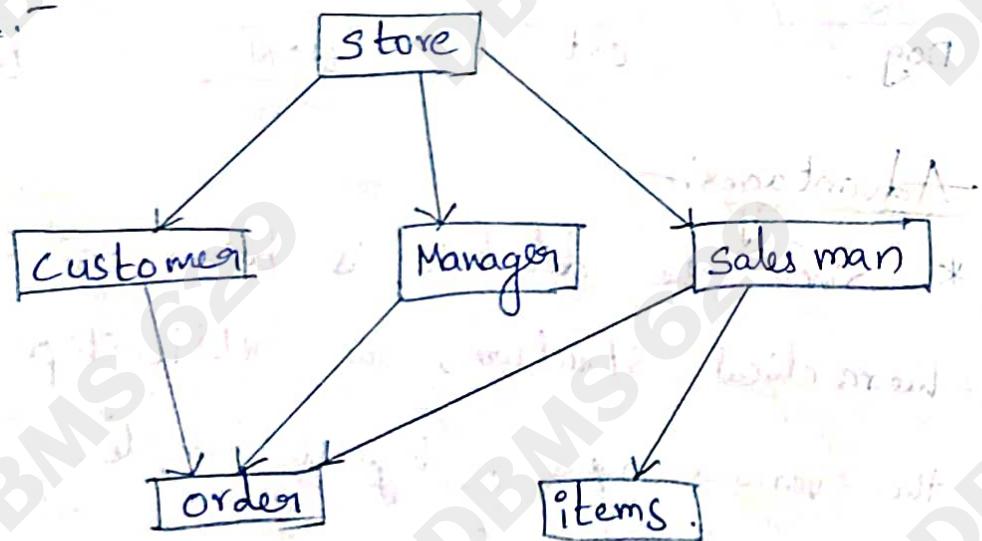
Disadvantages :-

- * Although it is conceptually simple and easy to design, its quite complex to implement.
- * If you make any change in the database structure, then it required to make necessary changes in all application programs that access the database.
- * It cannot handle many-to-many relationships.

2. What is Network data Model?

- A. Network data model has the entities which are organised in a graphical representation and some entities in the graph can be accessed through several paths.

Ex:-



Advantages:-

- * The network model is conceptually simple and easy to design.
- * supports many - to - many relationships.
- * Data access is flexible.
- * Data independence : change in data characteristics do not require change to application program.

Disadvantages:-

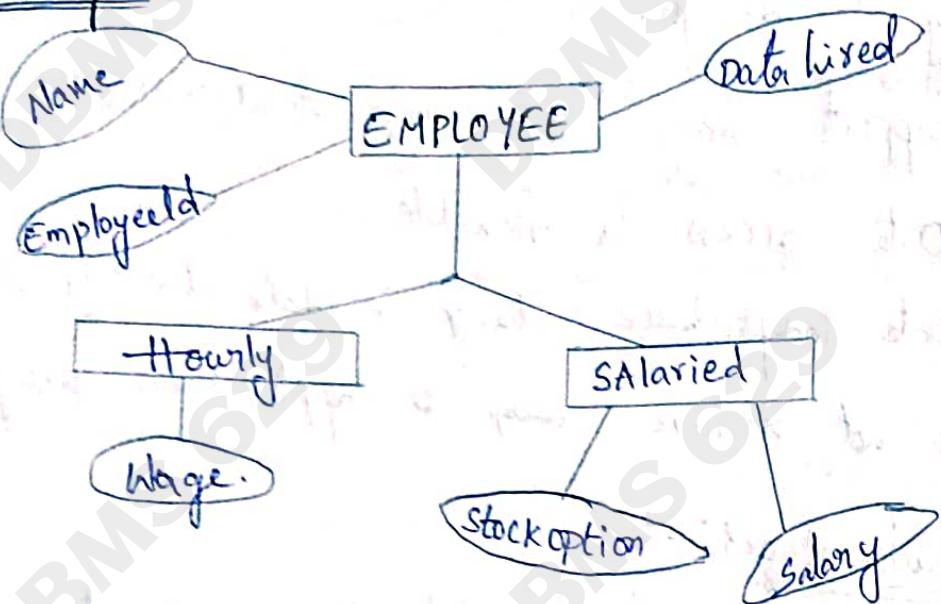
- * Network databases are difficult to design and use properly.
- * It is difficult to make changes in a database.
- * It gives very complex structure for the application programmer.
- * Not user friendly.

3. What is Relation Data Model?

- A. It is the most popular model and the most extensively used model. In this model the data can be stored in the tables and this storing is called as relation, the relations can be normalized and the normalized relation values are called atomic values. Each row in a relation contains unique value and it is called

as tuple, each column contains values from same domain and it is called ~~as~~ attribute.

Example:-



Advantages:-

1. Easy to use.
2. Flexibility:- Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.
3. Precision:- The usage of relational algebra and relational calculus in the manipulation of the relations between the tables ensures that there is no ambiguity.
4. security:- security control and authorized can also be implemented more easily.

5. Data Independence :-

Data Independence is achieved more easily than in the more complicated tree or network structure.

6. Data Manipulation Language :- With DML the transactions can be made very easily.

D's advantages :-

- * Needs powerful hardware.

What is DDL?

DDL is a short name of Data definition language which deals with database schemas and descriptions, of how the data should reside in the data base. In this DDL some commands are there. They are Create, Alter, Drop, Truncate, Comment, Rename etc.

Create :- To create a data base and its objects like [table, index, procedure].

Alter :- Alter the structure of the existing database.

Drop :- It is used for delete objects from the data base or table.

Truncate :- Remove all records from a table, including all spaces allocated for the records are removed.

Comment :- Add comments to the data dictionary.

Rename :- Rename an object.

Syntax :-

create table <filename> {
column-name1 data-type,
column-name2 data-type,
:
column-name n data-type,
};

Ex :- Create table books

create table employee
{
employee id int;
employee name char;
age int;
city char[25];
phno. varchar;
};

2. What is DML?

A. DML is a data manipulation language which deals with data manipulation and includes most common SQL statements such as SELECT, INSERT, UPDATE, DELETE.

SELECT:- It is used for to retrieve the data from a data base.

Insert:- It is used for to insert data in to a table.

UPDATE:- Updates existing data within a table.

Delete:- Delete all records from a data base table.

Merge:- UPSERT OPERATION [Insert (or) update].

call:- call a SQL / PL (or) Java sub program.

Explain plan:- Interpretation of the data access path.

Lock Table:- Concurrency control.

Syntax:-

SELECT attribute-list

FROM table-list

<WHERE condition>

Example:-

① Select * from stud;

② select * from stud where roll no = 1;

③ select NAME, Address from stud;

④ select name, Address from stud where rollno=1

⑤ select Name from stud where address

= 'Nashik';

3. What is TCL?

A. TCL is a transaction control language

which deals with a data base

In this language some commands are there.

They are * COMMIT * ROLL BACK

* SAVE POINT.

COMMIT :- COMMITS a transaction

ROLLBACK :- Roll back a transaction in case of any error occurs.

SAVE POINT :- To roll back the transaction

making points with in groups.

4. What is DCL?

A. DCL is the data control language which

includes commands such as Grant and mostly concerned with rights, permissions and other controls of the data base system.

Grant :- Allow users access privileges to the database.

Revoke :- Withdraw a user access privileges given by a grant command.

Language	Command List
DDL	Create Alter Drop Truncate / RENAME
DML	Select Insert Update Delete
DCL	Grant Revoke
TCL	commit Roll back Save Point set Transaction

1. What is primary key?
- A. Primary key is a special relational database table column designed to uniquely identify all the table records.

* A primary key's main features are:-

- It must contain a unique value for each row of data
- It cannot contain null values.

A primary key is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

Example:-

Students are routinely assigned unique identification (ID) numbers, and all adults receive government-assigned and uniquely identifiable social security numbers.

2. What is candidate key?

A. * A candidate key is the most minimal subset of few fields that uniquely identifies a tuple.

* Candidate keys are defined as the set of fields from which primary key can be selected.
It is an attribute or set of attribute that can act as a primary key for a table to uniquely identify each record in that table.

Ex:- Let's take an example of an employee table:-

CREATE TABLE EMPLOYEE (

Employee ID,

Employee Name,

SSN,

Dept ID,

).
Here in employee table employeeID & SSN are eligible for a primary key and thus are Candidate key.

3. What is Super key?

A. A super key is a combination of columns that uniquely identifies any row within a relational database management system (RDBMS) table. A candidate key is closely related concept where the super key is reduced to minimum number of columns required to uniquely identify each row.

4. What is foreign key?

A. * A foreign key is a field in one table that uniquely identifies a row of another table. In simpler words, the foreign key is defined in a second table, but it refers to the primary key in the first table.

* For example, a table called employee has a primary key called employee-id. Another table called employee details has a foreign key which references employee-id in order to uniquely identify the relationship between both the tables.

* A foreign key is a column or group of columns in a relational database table that provides the link between data in two tables. It acts as a cross references between tables because it references the primary key of another table, thereby establishing a link between them.

Employee.

Employee - ID	Employee - Name
1	John
2	Archer
3	Morgan
4	stonis

Employee Details.

Employee - ID - ref	city	Town(Native)	Pincode.
1	London	Spain	100243
2	Banglore	Mysore	560021
3	Hyderabad	Vijaywada	432002
4	Kolkata	Hyderabad	300642.

What is relationship?

A relationship is an association between several entities. For example, we could define a relationship

between an account and a customer indicating that the particular account belongs to the particular customer.

What is relationship sets?

A relationship set is a set of relationships of the same type. For example students studying in college

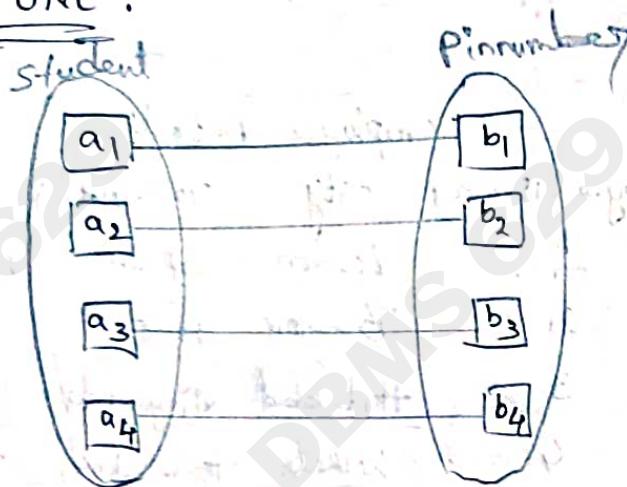
is one relation and students staying in hostel is another relation as both the relation belong to

Students they form student relationship set.

Mapping constraints :-

Mapping constraints express the number of entities to which another entity can be associated via relationship set.

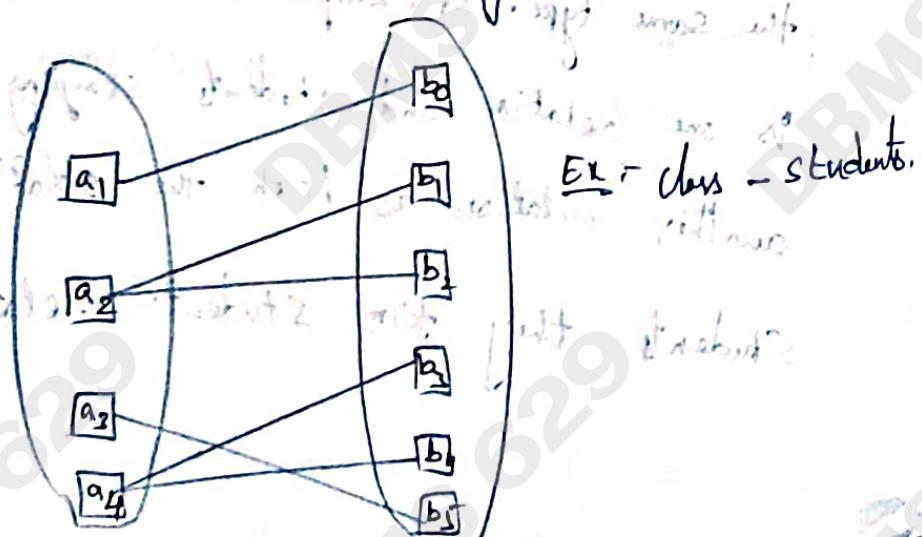
ONE - TO - ONE .



An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A, as shown in the above figure. Ex:- student — pin-no.

ONE - TO - MANY :-

An entity in A is associated with any number of entities in B. An entity in B, however can be associated with at most one entity in A.

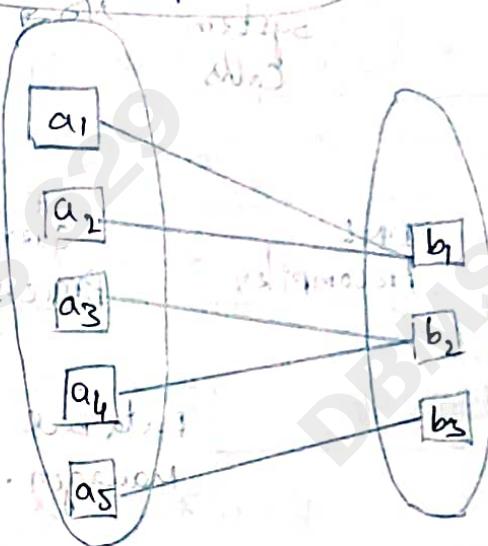


Ex - class - students.

Many - To - ONE

An entity in A is associated with at most one entity in B. An entity in B, however can be associated with any number of entities in A.

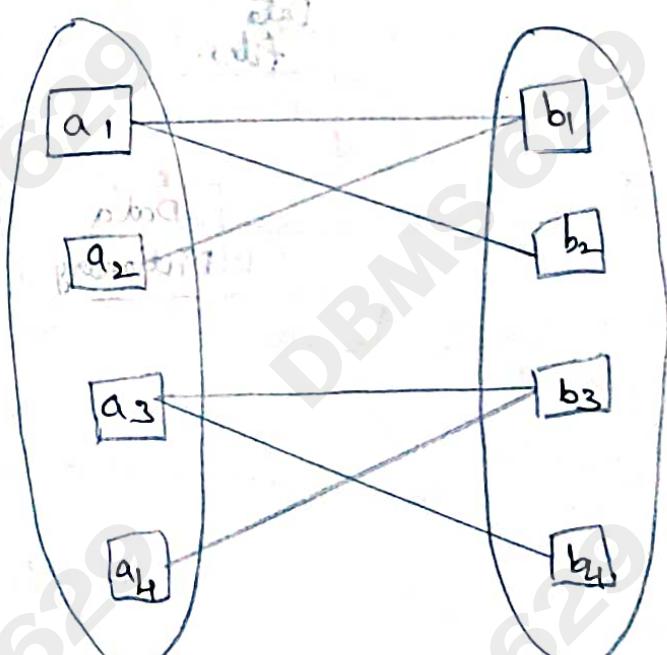
Ex :- Books → publisher



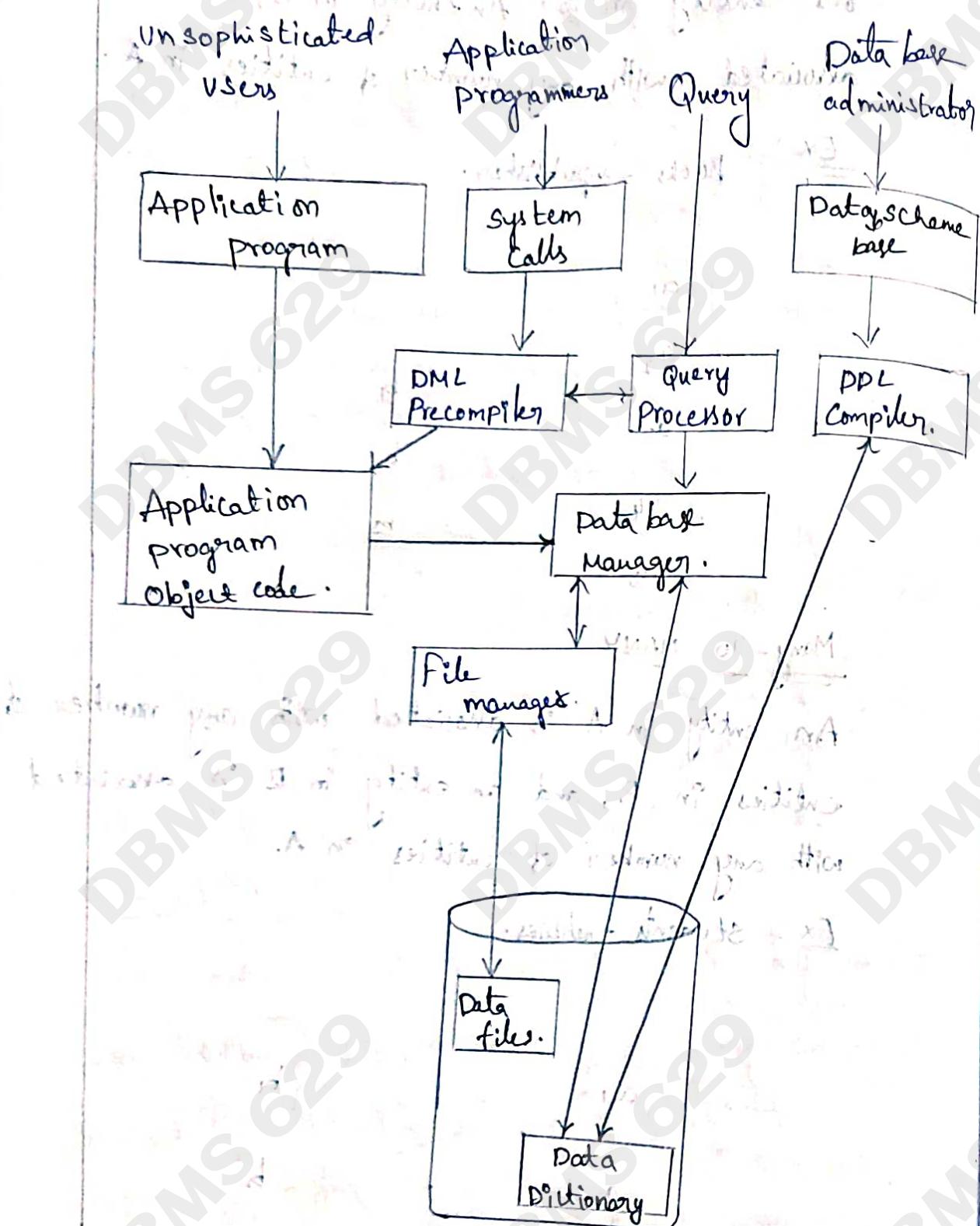
Many - To - MANY

An entity in A is associated with any number of entities in B, and an entity in B is associated with any number of entities in A.

Ex :- students - hobbies.



OVERALL SYSTEM STRUCTURE



Explain about data Models?

- A. * A data model is a collection of concepts, that can be used to describe the structure of the data base.
- * Data models include a set of basic operations, for specifying retrieval and update on the database.
- * Data models are a collection of conceptual tools for describing data, data relationships, data semantics and data constraints. There are three different groups.
1. Object - based Logical models.
 2. Record - based logical models.
 3. Physical Data Models.

what is Logical Data Independence?

- A. * The ability to modify the conceptual scheme without causing application programs to be rewritten.
- * usually done when logical structure of database is altered.
- * Logical data independence is harder to achieve as the application programs are usually heavily dependent on the logical structure of the data.

Explain about Entity & Entity set?

- A. The fundamental item in any data model is the entity. Entity can be defined as anything which can be distinctly identified. A place, person, picture, thing etc. Entities are classified can be classified into
- * Weak entity
 - * strong entity (or) Regular entity.

Weak entity:

A weak entity is an entity that is existence dependent on some other entity.

Ex:- An employees dependents.

Strong Entity:-

A regular (or) strong entity on the other hand, is an entity that is not weak.

Ex:- Employees.

Entity sets:- An entity set is a set of entities of the same type.

* Entity sets need not be disjoint.

Define GENERALIZATION, SPECIALIZATION & AGGREGATION

- A. GENERALIZATION: Generalization is the result of taking the union of two or more different entity sets to produce higher level entity set. The

attribute of higher level entity sets are inherited from lower entity sets.

SPECIALIZATION —

Specialization is the reversal of generalization.
Specialization is the process of dividing entity sets into lower level entity sets.

AGGREGATION :-

Aggregation is the technique of representing relationships among relationships is known as aggregation. One drawback of E-R diagram is that it is not possible to express multiple relationships among entities properly.

Explain about Normalization & BCNF?

A. Normalization is a process of decomposing complex tables into small tables.

* Normalization

* Normalization is the process of organizing data in a database. There are three types of normalizations

and there they are * 1st Normal form

* 2nd Normal form

* 3rd Normal form

* BCNF (Boyce - codd Normal form).

The following terms may be useful in normalization process:

TERM.	MEANING.
Relation.	A table.
Tuple	A row on a grid in a relation
Attribute	A field or a column in a relation
Cardinality of a relation.	The number of tuples in a relation.
Degree of a relation.	The number of attributes in a relation.
Domain of an attribute	The set of all values that can be taken by the attribute
primary key.	A primary key can be defined as a set of columns used to uniquely identify rows of a table.
Composite primary key.	In some tables a single column cannot be used to uniquely identify entities. In that case we have to use two or more columns to uniquely identify rows of the table. When a primary key contains two or more columns it is called as composite primary key.

1st Normal Form:-

A relation R is in first Normal form if and only if underlying domains of the relation contain atomic values.

This definition can be expressed in the two forms:-

1. * Every tuple of the Relation R, no attribute should have repeating groups.
2. * In every tuple of the relation R, each attribute must have a value and that too an atomic (indivisible) value.

Ex:- Student table.

student No.	NAME	subjects .
202	Prakash	Maths
202	Prakash	Chemistry
202	Prakash.	Physics.
209	Paramesh	Physics
209	Paramesh	BCE
209.	Paramesh	Maths.

2nd Normal form:-

A relation is said to be in 2nd normal form if and only if it is in 1st normal form and every non key attribute is fully dependent on the primary key.

Ex:-

Faculty:

F - Name	Subject
Prakash	Chemistry
Paramesh	BCE

child:

Child name	Child age -	Child sex .
Venkatesh.	18	M
monica	19	F

IIIrd Normal Form:-

A relation R is said to be in third Normal form if and only if it is in 2NF and every non key attribute is non-transitively dependent on the primary key.

Ex:-

Teacher - subject :-

Teacher name	Subject
Hari	DBMS
Surendra	DS
Dinesh	DE

~~Subject - Period.~~

Subject ref	Period.
DBMS	1,2
DS	3
DE	4

BCNF (Boyce - Codd Normal Form) :-

A relation is in BCNF if it is in 3NF and all of its determinants are candidate keys. For most practical purpose 3NF and BCNF have the same effect.

Ex:- Student-allotment - 1

stu-id	class code	Grade
125	phy	A
125	chem	C
128	Chm	B
238	phy	C
238	Chem	B

Student allotment - 2.

	Class code	Teacher #
Pranay	Phy	25
A. Asstt.	Chem	20
Shivam	Chm	20
Pranay	phy	25
Pranay	Chem	20

Explain E.F. CODD's Rules for RDBMS!

A. Rule 0:- Foundation Rule.

Rule 1:- Rule of Information.

Rule 2:- Rule of Guaranteed Access.

Rule 3:- Rule of Systematic Null Value.

Applies to every tuple in every relation.

Rule 4:- Rule of active and online relational catalog.

Rule 5:- Rule of comprehensive Data sub language.

Rule 6:- Rule of updating views.

Rule 7:- Rule of set level insertion, update, and deletion.

Rule 8:- Rule of physical data independence.

Rule 9:- Rule of logical data independence.

Rule 10:- Rule of integrity independence.

Rule 11:- Rule of distribution independence.

Rule 12:- Rule of Non Subversion.

* Define instances?

A database instance is a set of memory structures that manages database files. A database ~~files~~ is a set of physical files on disk created by the CREATE DATABASE statement. The instance manages its associated data and servers the users of the database.

* Define Schema?

A schema is the organisation and structure of a database. A schema contains schema objects, which could be tables, columns, datatypes, views, stored procedures, relationships, primary keys

foreign keys, etc.,

A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

List some schema objects?

Include tables, views, sequences, synonyms, indexes, clusters, database links, procedures and packages.

What are the responsibilities of Database Manager?

- * It is the interface between low-level data, application programs and queries.
- * Database typically require lot of storage space. This must be stored on disks.
- * The goal of the database system is to simplify and easy to access the data.

Responsibilities:-

1. Interaction with the filer manager:-

* storing raw data on disk using the file system usually provided by operating system.

* The database manager must translate DML statements in to low-level file system commands.

2. Integrity Enforcement:-

* Checking that updates in the database do not violate consistency constraints.

3. Security Enforcement

Ensuring that users only have access to information they are permitted to see.

4. Backup and Recovery

Detecting failures due to power failure, disk crash, software errors, etc and restoring the database to state before the failure.

5. Concurrency Control

Preserving data consistency when there are concurrent users.

Define DB:-

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system.

2. CONCEPTS OF SQL

Define SQL?

- A. Structured Query Language (SQL) is a programming language that is typically used in relational database or data stream management system. It was developed by IBM in the early 1970s and now an official standard recognized by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO).

List out the SQL features?

- A. * SQL stands for structured query language.
* SQL allows you to access a database.
* SQL is an ANSI standard computer language.
* SQL can execute queries against a database.
* SQL can retrieve data from a database.
* SQL can insert new records in a database.
* SQL can delete records from a database.
* SQL can update records in a database.

List out benefits of SQL?

A. The benefits of SQL are:-

- * SQL is an English like language.
- * Easy to learn.
- * Application portability.

- * More productivity with a single statement we can perform task like creation of table etc.
- * Cross system communication.
- * SQL standard exists. All RDBMS support SQL.
- * Both interactive and embedded access.
- * can be used by specialist and non-specialist.
- * SQL provides automatic navigation to data.

Write a few lines about Embedded SQL?

A This is a method for combining data manipulation capabilities of SQL and computing power of any programming language. Then embedded statements are in line with the program source code of the host language. The code of embedded SQL is parsed by a preprocessor which is also embedded and is replaced by the host language called for the code library it is then compiled via the compiler of the host. Most popular hosting language is c, it is called for example pro*c in Oracle and sybase database management systems and ECPG in the progressql data base management system.

Example:-

```
int age;
char accno[6];
:
scanf ("%d", &age);
scanf ("%s", accno);
```

EXEC SQL update ACCOUNT

SET age=:age where accno=:accno;

Explain about lexical conventions?

A. SQL statements are composed of commands, variables, and operators, which are described in detail in this and subsequent chapters. A SQL statement is constructed from:

- * character A through Z.
- * Numbers 0 through 9
- * spaces.
- * The following special characters: +, -, *, =, ?, !, @, (,), -, ., , <, >, /, \$, #

A SQL statement can contain one or more of the following items anywhere a single space can occur:-

- * Tab
- * Carriage return
- * Multiple spaces.
- * comments.

What are the rules of naming of SQL?

- A. *
- * They may comprise 1 to 30 alphanumeric characters.
 - * They must begin with a letter.
 - * They may include an underscore (-).
 - * They may include a dollar (\$), pound sign (#), although Oracle discourages the use of these characters.
 - * They may not be a reserved word.
 - * They may not be the name of a SQL command.

Explain about above the objects and parts?

A. A schema object is a logical collection of data or other objects that are owned by a user and stored in the database. The following types of objects are considered schema objects.

- * Clusters
- * Database links
- * Database triggers.
- * Indexes.

Part of objects:-

Some objects are made up of parts that you must also name, such as:

- * Columns in a table or view.
- * Integrity constraints on a table.
- * Packaged procedures, Packaged stored functions, and other objects stored within a package.

Write about Literals text and integers?

A. Literal is fixed data value. There are four kinds of literals supported in SQL. They are:-

- * Text or character literal.
- * Numeric literal
- * Bit strings
- * Date and Time literals.

* Text or character :-

Text or character literals are written as sequence of strings written as a sequence of character enclosed in single quotes.

Example :-

'structured Query Language'

'govinda naidu'

'18101 - CM - 235'

* BIT STRING:-

A bit string is written either as a sequence of 0s and 1s enclosed in single quotes and preceded by the letter 'B' or as sequence of Hexadecimal digits enclosed in single quotes and preceded by letter 'x'.

Example :-

B'101101'

B'1'

x'A5'

* Numeric Literals :- These are used to construct exact numbers and approximate numbers. Syntax rules of numeric literals are :-

* A numerical literal can be written in signed integer form, signed real numbers without exponents, or real numbers with exponents.

Examples:-

1
-22

33.3

-44.4

55.55e5

-666.66e-6.

Date & Time literals:-

They are used to construct date & time values.

The syntax of date and time literals are:-

* The date literal is written in the form of
"DATE 'yyyy-mm-dd'".

* A time literal is written in the form of
"TIMESTAMP 'yyyy-mm-dd hh:mm:ss'".

write a brief note on PSEUDO column?

* Null is the absence of data; it is neither character nor numeric. Both character and numeric data elements can be set to NULL, which indicates that the element contains no value whatsoever.

* Null is not the same as zero. In fact, any arithmetic operation involving a data element

containing NULL will evaluate to NULL. For example, if the current value of a is NULL, then the following expression will evaluate to NULL:

$$(a + 10) * 20.$$

A pseudo column is an item of data which does not belong in any particular table but which can be treated as if it did. Any SELECT list of columns can include these pseudo columns.

SYSDATE current date and time

ROWNUM sequence number assigned to retrieved rows

ROWID unique identifier for a row

UID Number associated with a user

USER Userid of current user

SQL LOGICAL OPERATORS

A logical operator is used to produce a single result from combining the two separate conditions.

operator	action
AND	Returns true if conditions are true.
OR	Returns true if either of the condition is true
NOT	Returns if the condition is false.

SUB QUERY

A subquery (or) inner query (or) a nested query is a query with in another SQL query and embedded within the where clause.

* A subquery is used to return data that will be used in the main query as a condition of further restrict the data to be retrieved.

* It can be used with select, insert, update and delete statements.

Consider the customers table

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	35	Delhi	2000.0
2	Suresh	25	Delhi	6000.0
3	Pandu	42	Kolkata	7000.0
4	Hardik	29	Mumbai	8000.0
5	Sweety	19	Banglore	18000.0

Now, let check the following subquery with a Select

Select

SQL > select * from customers

Where ID IN (select ID from customers where
salary > 4500);

ID	NAME	AGE	ADDRESS	SALARY
2	Suresh	25	Delhi	6000.0
3	Pandu	42	Kolkata	7000.0
4	Hardik	29	Mumbai	8000.0
5	Sweety	19	Banglore	18000.0

Insertion :-

SQL > insert into customers - BKP

select * from customers

Where ID IN (select ID from customers).

Update :-

SQL > update customers

set salary = salary + 0.25

Where age IN (select age from customers - BKP

Where Age >= 27)

Delete :-

Delete from table - name
 where operator (value) to update or insert
 (select column - name from table - name)
 where

SQL & delete from customers
 (select age IN (select AGE from customers - BKP
 where age
 where age ≥ 27);

Single row sub Query :-

A single row subquery returns zero, or
 one row to the outer SQL statements. You can
 place a subquery in a WHERE clause, a HAVING
 clause, or a FROM clause of select statement.

AGENT-CODE	AGENT-NAME	WORKING-AREA	COMMISSION	PH-NO.
A 00 2	Dhananjay	Kolkata	0.29	08310692638
A 00 8	Rohit	Bangalore	0.4	07702432291
A 00 9	Pavan	Hyderabad	0.31	06363580263

Select Agent-name, agent-code, phone-no from Agents.

Where agent - code

(select agent_code

from agent

Where agent-name = Dhananjay;

and commission is less than 0.29

on constraint is made (1) AND

Leave prints form

SQL Functions

There are two types of SQL functions. They are aggregate functions and scalar (non-aggregate) functions. Aggregate function operate on many records and produce a summary, works with group by whereas non-aggregate functions operate on each record independently.

SQL Aggregate Functions:-

This function can produce a single value for an entire group or table. They operate on sets of rows and return results based on groups of rows. Some Aggregate functions are:-

* SQL COUNT function

* SQL SUM function

* SQL AVG function

* SQL MAX function

* SQL MIN function.

* SQL COUNT() Function:-

The SQL COUNT() function returns the number of rows in a table satisfying the criteria specified in the WHERE clause. It sets the number of rows of non NULL column values.

COUNT() returns 0 if there were no matching rows.

Syntax:-

COUNT(*)

↳ No. → COUNT([ALL|DISTINCT] expression),

Example:-

↳ Select * from tbl-employees.

↳ SELECT COUNT(*) FROM tbl-employees

Id	Emp-name	emp-age	emp-salary	Join-date
1	Raju	35	15000.00	2017-01-01 00:00:00.000
2	Hari	30	10000.00	2019-01-01 00:00:00.000
3	Reddy	27	17000.00	2014-08-29 00:00:00.000
4	Chaitanya	36	19000.00	2018-04-03 00:00:00.000
5	Ram	45	20000.00	2016-03-01 00:00:00.000
6	Sambal	29	19000.00	2015-09-23 00:00:00.000

* SQL SUM() Function :-
The SQL AGGREGATE SUM() function returns the sum of all selected columns.

Syntax :-
SUM([ALL|DISTINCT] expression)

Example:-

Select * from tbl-emp-salary-paid

SELECT SUM(emp-sal-paid) AS Total Paid

FROM tbl-emp-salary-paid.

emp-id	emp-name	emp-sal-paid	date-Paid
1 1	Raju	15000.00	2019 - 04 - 02
2 2	Hari	10000.00	2019 - 04 - 02
3 3	Reddy	17000.00	2019 - 04 - 02
4 4	Chaitanya	19000.00	2019 - 04 - 02
5 5	Ram	20000.00	2019 - 04 - 02
6 6	Samba	19000.00	2019 - 04 - 02

Total - paid

\$ 100,000.00

SQL AVG() function

SQL AVG() function calculates the average

value of a column of numeric type. It returns

the average of all non NULL values.

Syntax :- AVG([ALL | DISTINCT] expression)

Example :-

USE Company;

SELECT Education, AVG(Yearly_Income) AS Avg_Income
FROM customerdetails GROUP BY Education;

GROUP BY Education;

	Education	AVG(Yearly-Income)
1.	Bachelors	79000.00
2.	Database	95000.00
3.	Degree	87000.00
4.	Graduate degree	81000.00
5.	High school	56000.00
6.	Masters	120000.00
7.	Post Graduate	542500.00
8.	Under Graduate	79000.00
9.	Profit	55000.00

JOINS

An SQL join clause - corresponding to a join operation in relational algebra combines columns from one or more tables in relational database. It creates a set that can be saved as a table or used as it is. A JOIN is a means for combining columns from one (self-join) or more tables by using values common to each.

SQL> desc employee;

Name	Type
EMP ID	NUMBER(3)
EMP NAME	VARCHAR2(15)
JOB	VARCHAR2(10)
SALARY	NUMBER(8,2)
ADDRESS	VARCHAR2(15)

SQL > Select * from employee;

EMPID	EMPNAME	JOB	SALARY	ADDRESS
1	Abhishek	Developer	20000	Hyderabad
2	Raju	Driver	8000	Hyderabad
3	Ram	Developer	13000	Banglore
4	Sowmya	Designer	18000	Banglore
5	Schandhana	Testing	17000	Amerpet
6	Sravani	Teacher	7000	Secundabad
7	Arpana	Teacher	7000	Oldcity
8	Srikanth.	Teacher.	7000	Hitechcity

8 rows selected.

SQL > desc emp;

Name of primary key
1 ID is a primary key and also it is auto increment type NUMBER(3).
GENDER is VARCHAR(6).
PHNO is NUMBER(10).
EXPERIENCE is NUMBER(2).

SQL > Select * from emp;

ID	GENDER	PHNO	EXPERIENCE
1	Male	8310692638	2
2	Male	770243291	5
3	Male	6363767693	1
4	Female	724802370	1
5	Female	8363520263	2
6	Female	7942354220	0

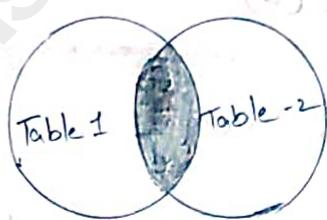
6 rows selected.

Inner Join :-

The INNER JOIN keyword selects records that have matching values in both tables.

Syntax:- (s allst) select table1, table2 from table1 inner join table2;

```
SELECT column-name(s)
FROM table1
INNER JOIN table2
ON table1.column-name = table2.column-name;
```



[INNER JOIN]

SQL> select empname, job, gender, experience
from employee, emp
where employee.empid = emp.id;

EMPNAME	JOB	GENDER	EXPERIENCE
Abhishek	Developer	Male	2
Raju	Driver	Male	5
Ram	Developer	Male	1
Sowmya	Designer	Female	3
Chandana	Testing	Female	2
	Testing	Female	most

LEFT JOIN:-

The LEFT JOIN keyword returns all records from the left table (table 1), and the matched records from the right table (table 2). The result is NULL from the right side, if there is no match.

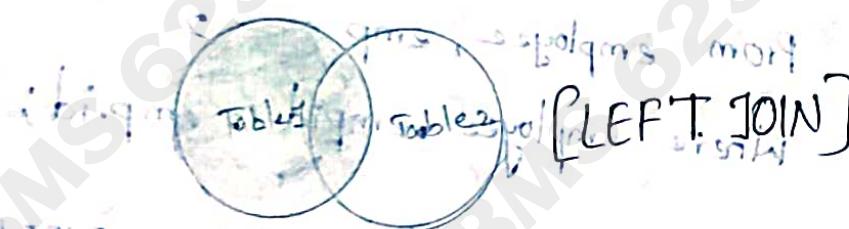
Syntax:-

SELECT column_name(s)

FROM table1

LEFT JOIN table2

ON table1.column_name = table2.column_name;



SQL > select empname, job, gender, experience

from employee leftjoin emp ON
employee.empid = emp.id;

EMPNAME	JOB	GENDER	Experience
Abhishek	Developer	Male	2 years
Raju	Driver	Male	5 years
Ram	Developer	Male	1 month
Sowmya	Designer	Female	1
Chandana	Testing	Female	2
Sravani	Teacher		
Arpana	Teacher		
Srikanth	Teacher		

8 rows selected

RIGHT JOIN :-

The RIGHT JOIN keyword returns all the records from the right table (Table 2), and the matched records from the left table (Table 1). The result is NULL from the left side, when there is no match.

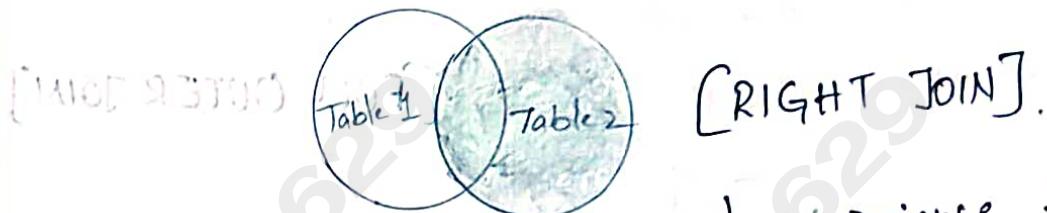
Syntax :-

SELECT column-name(s)

FROM table1

RIGHT JOIN table2

ON table1.column name = table2.column name;



SQL > select empname, job, gender, experience from

employee full join emp ON

employee.empid = empid ;

EMPNAME	JOB	GENDER	EXPERIENCE
Abhishek	Developer	Male	2
Raju	Driver	Male	5
Ram	Developer	Male	1
Sowmya	Designer	Female	1
Chandana	Testing	Female	2
Sravani	Teacher		
Aprana	Teacher		
Srikanth	Teacher	Female	0

9 rows selected.

FULL OUTER JOIN

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records. The FULL OUTER JOIN and FULL JOIN are the same.

Syntax:-

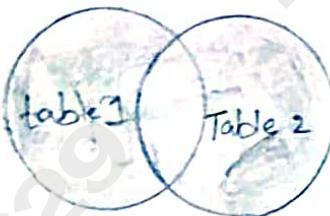
SELECT column_name(s)

FROM table1

FULL OUTER JOIN table2

ON table1.column-name = table2.column-name

WHERE condition;



[FULL OUTER JOIN]

ANTI JOIN:-

An antijoin returns rows from the left side of the predicate for which there are no corresponding rows on the right side of the predicate. That is, it returns rows that fail to match (NOT IN) the subquery on the right side.

The following example selects a list of employees who are not in particular set of departments:

SELECT * FROM employees

WHERE department_id NOT IN (

(SELECT department_id FROM departments

WHERE location_id = 1700);

SEMI JOINS :-

A Semijoin returns rows that match an EXISTS subquery without duplicating rows from the left side of the predicate when multiple rows on the right side satisfy the criteria of the subquery.

Semijoin and antijoin transformation cannot be done if the subquery is on an OR branch of the WHERE clause.

Ex:- SELECT * FROM departments
 WHERE EXISTS
 (SELECT * FROM Employees
 WHERE departments.department-id =
 employees.empdepartment-id
 AND employees.Salary > 2500);

SQL: MIN() And MAX():

The MIN() function returns the smallest value of the selected column.

MIN() syntax

SELECT MIN(column-name)

FROM table-name

Where condition;

productID ProductName supplierID categoryID quantity Price

1	Tomato sauce	1	1	10 boxes x 2 bags	12
2.	soya sauce	1	1	24-12 bottles	19
3	strawberry syrup	1	2	12 - 550 ml bottles	10
4.	Mango syrup.	2	2	48-6 jars	22
5	Rose syrup.	2	2	36 boxes.	21.35

Example:-

SELECT MIN(price) AS
smallest price
FROM products;

Output

smallest price

2.5

MAX() Syntax

SELECT MAX(column-name) FROM table-name AS largest price
WHERE condition;

Example

SELECT MAX(price)!

As largest price

FROM products;

Output

Largest price

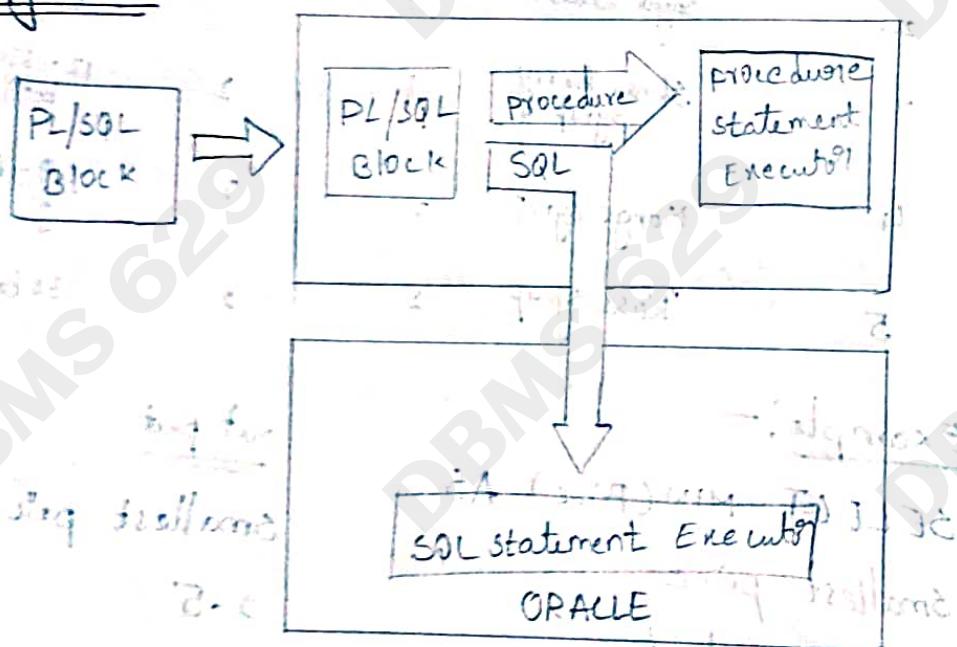
263.5

CHAPTER-3

What is PL/SQL?

PL/SQL is oracle corporation's procedural extension for SQL and the oracle relational database. PL/SQL is available in oracle database, Times ten in-memory database, and IBM DB.

Diagram :-



Advantages:-

1. Better performance, as SQL is executed in bulk rather than a single statement.
2. High productivity.
3. Tight integration with SQL.
4. Full portability.
5. Tight security.

6. supports object oriented programming concepts.

7. scalability and management.

8. supports web Application Development.

9. supports server page Development.

Disadvantages:

1. stored procedures in PL/SQL uses high memory.

2. Lacks functionality debugging in stored procedures.

3. Any change in underlying database requires

change in the presentation layer also.

4. Does not completely separate roles of back-end developer and front-end developer.

5. Difficult to separate HTML development with PL/SQL development.

What is PL/SQL Loops?

A In PL/SQL we have three different loop options to choose from when we want to execute a statement repeatedly in our code block.

They are :- * Loop Statement

* While Loop Statement

i(i) until For loop.

i(i) = i

i(i) <= i

(GAS)

Loop Statement

- * Simple loop always begins with the keyword LOOP and ends with a keyword END LOOP.
- * A basic / simple loop can be terminated at any given point by using the exit statement.
or by specifying certain condition by using the statement exit when.

Syntax :-

LOOP [do repeat iteration] with [repeat sequence of statements] until [repeat condition for loop]

END LOOP;

Example:-

set serveroutput on;

DECLARE

i int;

begin

for i:=1 to 10 loop

loop

if i>10 then

exit;

end if;

dbms_output.put_line(i);

i:=i+1;

end loop;

end;

While loop Statement:-

It is an entry controlled loop which means that before entering in a while loop first the condition is tested, if the condition is TRUE the statements in a group of statements get executed and if the condition is FALSE the control will move out of the while loop.

Syntax:- WHILE < test - condition > LOOP

< action >

END LOOP;

(i) until loop, loops - controls

Example:-

set serveroutput on;

```
DECLARE
    num int := 1;
BEGIN
    while (num <= 10) LOOP
        dbms_output.put_line("|| no");
        num := num + 2;
    end loop;
END.
```

For Loop:- It is an entry controlled loop. The for loop automatically increments the value of the counter variable by 1 at the end of each loop cycle.

Syntax:-

FOR counter_variable IN start_value .. end_value Loop
 with loop condition if true, then it will execute
 Statement to be executed for stated loop
 END LOOP;

Example:-

```
set serveroutput on;
DECLARE
  i number(2); -- loop variable
BEGIN
  FOR i IN 1..100 LOOP
    dbms_output.put_line(i);
  END LOOP;
END;
```

What is conditional statements.

- A. Decision making statements are those statements which are in charge of executing a statement out of multiple given statements based on some condition. The condition will return either true or false. Based on what the condition returns, the associated statement is executed.

The decision making statements in PL/SQL are 4 types. They are:

* if then statement

If it's then; else statement

or nested if - then statements

if - then - else if - then - else ladder.

If then statement:— If then statement is the most simple decision-making statement. It is used

to decide whether a certain statement or block of statements will be executed or not i.e. if a certain condition is true then a block of statement is executed otherwise not.

Syntax:

if condition then

-- if do something --

end if;

Example:

Declare

num1 number := 10;

num2 number := 20;

begin

if num1 > num2 then

dbms-output . put-line ("num1 is small")

end if;

((*) dbms-output . iput-line ("I am Not in if"))

if end;

((*) dbms-output . iput-line ("if has"))

2. If - then - else

The if statement alone tells us that if a condition is true it will execute a block of statements, and if the condition is false it won't. But what if we want to do something else if the condition is false? Here comes the else statement. We can use the else statement with if statement to execute a block of code when the condition is false.

Syntax:-

if (condition) then {
 // block of code
}

--- Executes this block if..

--- condition is true.

else

 --- Executes this block if

 --- condition is false

Example:-

declare

 num1 number := 10;

 num2 number := 20;

begin

 if num1 < num2 then

 dbms-output.put-line("I am in its block");

Else

 dbms-output.put-line("I am in else block");

end if

dbms - output . put-line ('in' are not in it or else
for Errmsg Name Journal wait - bug . bugnum - 2mabk
end.

3. Nested - if - then.

A nested if - then is an if statement that is the target of another if another if statement.

Nested if - then statements mean an if statement

inside another if statement. PL/SQL allows us to

nest if statements within if - then statements.

Syntax: if to we do need ch . much got

if (condition) then

it executes when condition1 is true will do

and to with if (condition2) then

it executes when condition2 is true here

end if;

end if;

Example:

declare

num1 number := 10;

num2 number := 20;

num3 number := 20;

begin

if num1 < num2 then

dbms - output . put - line ('num1 small num2');

if (num1 < num2) then . . .
dbms_output.put_line('num1 small num3 also');
end if;

end if;
dbms_output.put_line('after end if');

end;

4. If - then - else if - then - else ladder.

Here user can decide among multiple options.

The if then statements are executed from the top down. As soon as one of the conditions

controlling the if is true, the statement is associated with that if is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.

Syntax:-

if (condition) then

-- statement

elseif (condition) then

-- statement

else

-- statement

end if

Example:-

```
declare
    num1 number := 10; --num1 is initialized to 10
    num2 number := 20; --num2 is initialized to 20
begin
    if num1 < num2 then
        dbms_output.put_line('num1 small');
    elsif num1 = num2 then
        dbms_output.put_line('num1, num2 both equal');
    else
        dbms_output.put_line('num2 greater');
    end if;
    dbms_output.put_line("after end if");
end;
```

PL/SQL Tables:- PL/SQL tables are:

PL/SQL's way of providing arrays. Arrays are like temporary tables. In memory thus are processed very quickly. It is important for you to realize that they are not database tables, and DML statements cannot be issued against them. This type of table is indexed by a binary integer counter.

Why use PL/SQL tables?

- PL/SQL tables help you move bulk data. They can store columns or rows of oracle data, and they can be passed as parameters. So, PL/SQL tables make it easy to move collections of data into and out of database tables or between client-side application and stored subprograms.

Advantages of Exception Handling

- * provision to complete program execution
- * propagation of errors
- * identifying error types
- * meaningful error reporting
- * easy identification of program code through error handling code.

IN & OUT Program.

Declare

```
i number;  
j number;  
k number;
```

```
PROCEDURE findadd (num1 IN number, num2 IN  
Number, sum OUT number) IS  
(num1 || i and j) and k. begin. endb  
BEGIN  
    sum := num1 + num2;  
END;  
BEGIN  
    i := 5;  
    j := 5;  
    findadd(i, j, k);  
    dbms_output.put_line ('The sum is ' || k);  
END;  
/.
```

Expected Output :-

The sum is 10

INOUT program:

DECLARE

num number;

PROCEDURE addnum (i: INOUT number)

IS

BEGIN

i := i + i;

END;

BEGIN

num := 1; i := num;

addnum(num);

dbms_output.put_line('The sum is ' || num);

END;

1.

Expected output:

The sum is 2.

FUNCTION program:

DECLARE num out NUMBER; a number;

b number;

c number;

FUNCTION findmax (x IN number, y IN number)

RETURN number

IS

z number;

```

BEGIN
    IF x > y Then
        z := x;
    ELSE
        z := y;
    END IF;
    RETURN z;
END;

BEGIN
    a := 23;
    b := 45;
    c := findmax(a, b);
    dbms-output.putline('maximum number || Findmax');
    dbms-output.putline('Findmax');
END;

```

1.

~~Expected~~ Output :-

Maximum number 45;

RECURSION Program :-

```

DECLARE
    num number;
    factorial number;
Function fact(x number)
RETURN number
IS
    f number;

```

BEGIN

IF $x = 0$ Then

$f := 1;$

ELSE

$f := x * \text{fact}(x-1);$

END IF;

RETURN $f;$

END;

BEGIN

$\text{num} := 6$

~~number will be used for output~~ factorial := fact(num);

dbms_output.put_line('Factorial number

is || fact');

END;

1.

Expected OUTPUT: Factorial number is 720.

Global & LOCAL

Declare

num1 number := 95;

num2 number := 85;

BEGIN

629

```
dbms_output.put_line ('Outer variable number');
dbms_output.put_line ('Inner variable number');

Declare
    num1 number := 195;
    num2 number := 185;

BEGIN
    dbms_output.put_line ('Inner variable num1 number');
    dbms_output.put_line ('Inner variable num2 number');
END;
```

END;

1.

Expected output:

Outer Variable num1 is 195
Outer Variable num2 is 185
Inner Variable num1 is 195
Inner Variable num2 is 185

PL/SQL procedure successfully created.

Explain briefly about data types in PL/SQL?

A. datatype:- Data types are means to identify the type of data, and associated operators

• Example :- Shows the for handling it.

PL/SQL provides a variety of predefined data types. They are:

- * scalar type
- * composite type
- * reference type
- *LOB type.

* Scalar Data types:-

The scalar types are types that store single value such as number, Boolean, character, and char, varchar2 etc...

Char	Fixed length character
Varchar2	Variable length character.
Number.	Fixed decimal, floating decimal, or integer values, many subtypes are also available.
Boolean.	Logical data type, used for TRUE, FALSE values.
Date	Stores date and time information.
Long	character data of variable length upto 2GB
BINARY - INTEGER.	Signed integers (Older & slower version of PLS-Integer) magnitude range is - 2147483647 ... 2147483647
PLS - Integer.	Signed integers PLS-INTEGER values require less storage and provide better performance than NUMBER values. magnitude range is - 2147483647 ... 2147483647.

COMPOSITE DATA TYPES:-

Composite types have internal components that can be manipulated individually, such as the elements of an array, record or table.

REFERENCE DATA TYPE:

A reference type is a code object that is not stored directly where it is created, but that acts as a kind of pointer to a value stored elsewhere.

LOB (LARGE OBJECT) DATA TYPE:-

A LOB is used to store large data such as graphics, sound files, images. The maximum size of data is 4 GB.

* BLOBs (Binary LOBs) → used to store large data such as video clips, sound files etc.

* CLOBs [Character LOBs] → Similar to BLOB but stores single byte character data.

* NCLOBs [National character LOBs] → Used to store large blocks of character data from the National character set.

Data type conversion

Sometimes it is necessary to convert a value from one data type to another. PL/SQL supports both explicit and implicit (automatic) data type conversion.

Explicit conversion:

To convert values from one data type to another, you use built-in functions.

Implicit conversion

PL/SQL can convert the data type of a value automatically. This allows you to use A literals, variables, and parameters of some type where another type is expected.

Implicit conversions:

Table 1

CHAPTER-4

CONCEPTS OF NOSQL

Definition of NoSQL: As NOSQL (Not only SQL) database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. Motivations for this approach include simplicity of design, horizontal scaling, and finer control over availability. (OR)

A NOSQL database is a database that provides a mechanism to store and retrieve data other than the tabular relations used in relational databases.

The primary objective of a NOSQL database is to have, * Simplicity of design, and * Horizontal scaling and * Finer control over availability.

Advantages

• Simple
• Scalable

Disadvantages

• Inconsistent data

* Why Need for NoSQL?

- * Support large numbers of concurrent users (tens of thousands, perhaps millions).
- * Deliver highly responsive experiences to a globally distributed base of users.
- * Be always available with no downtime.
- * Handle semi-structured and unstructured data.
- * Rapidly adapt to changing requirements with frequent updates and changes, easily adding new features.

* Compare RDBMS & NoSQL

RDBMS	NoSQL
It stands for relational database management system.	It stands for not only structured query language.
Supports powerful query language.	Supports very simple query language.
It has a fixed schema.	No fixed schema.
Follows ACID (Atomicity, Consistency, Isolation, and Durability)	It is only "eventually consistent".
Supports transactions	Base properties
	Does not support transactions.

* What is CAP Theorem in NoSQL?

- A. In 2000, The CAP Theorem (or) Eric Brewer presented his keynote speech at the ACM Symposium on the principles of distributed computing and CAP theorem was born in States that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees.

* Consistency: Every read receives the most recent write or an error.

* Availability: Every request receives the most recent write without the guarantee that it contains the most recent write.

* Partition tolerance: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes.

When a network partition failures happens should we decide to

- * Cancel the operation and thus decrease the availability but ensure consistency.
- * Proceed with the operation and thus provide availability but risk inconsistency.

The CAP theorem implies that in the presence of a network partition one has to choose between consistency and availability.

Note that consistency as defined in the CAP theorem is quite different from the consistency guaranteed in ACID database transactions.

A — Atomicity

C — Consistency

I — Isolation

D — Durability

Atomicity:- Atomicity means transaction

either completes or fails entirely.

There is no state in between. No body sees a partial completion of a transaction.

Consistency:- Consistency means the transaction leaves the database in the valid state.

Isolation :- Isolation means no two transactions mingle & interface with each other. The result of two transactions executed in parallel would be same as sequential execution.

Durability :- Durability means the changes made by the transaction are saved & remains there even if power is turned off.

Explain About Base Properties?

A. Base stands for Basic Availability

Soft state Eventual Consistency

Basic Availability

The NoSQL database approach focuses on the availability of data even in the presence of multiple failures.

It achieves this by using a highly distributed approach to database management. Instead of maintaining a single large data store, and

focusing on the fault tolerance of that store,

NoSQL databases spread data across many storage systems with a high degree of replication.

In the unlikely event that a failure disrupts access to a segment of data, this does not

necessarily result in a complete database outage.

Soft State — BASE databases abandon the

consistency requirements of ACID model pretty much completely. One of the basic

concept behind BASE is that data consistency is the developer's problem and should not be handled by the database.

Eventual Consistency —

The only requirement that NoSQL databases have regarding consistency is to

require that at some point in the future, data will converge to a consistent state.

No guarantees are made, however, about when this will occur. That is a complete departure from the immediate consistency requirements of ACID that prohibits a transaction from executing until the prior transaction has completed and the database has converged to a consistent state.

For example, if two clients try to insert a new row into a table, the database must wait for both transactions to complete before it can return a response.

List Advantages of NoSQL

- A. 1. Can be used as primary or analytic data source.
2. Big data capability.
3. No single point of failure.
4. Easy Replication.
5. No need for separate caching layer.
6. It provides fast performance and horizontal scalability.
7. Can handle structured, semi-structured data with equal effect.
8. Object-oriented programming which is easy to use and flexible.
9. NoSQL databases don't need a dedicated high-performance server.
10. Support developer languages and platforms.
11. Simple to implement than using RDBMS.
12. It can serve as the primary data source for online applications.
13. Handling big data which manages data

velocity, variety, volume, and

complexity.

14. Excels at distributed databases and

multi-data center operations.

15. Eliminates the need for a

specific caching layer to store data.

16. Offers a flexible schema design which

can easily be altered without downtime

or service disruption.

* Differences Between ACID and BASE

ACID	BASE
strong consistency	weak consistency
Isolation	Availability first
Focus on "commit"	Best effort
Nested Transactions	Approximate answers [OK]
Availability?	optimistic (aggressive)
Pessimistic/conservative	Fast and simple
Difficult evolution (e.g. schema)	Easier evolution.

* List the disadvantages of NoSQL?

- A 1. NoSQL database don't have the reliability functions which relational databases have (basically don't support ACID).
2. This also means that NoSQL databases offer consistency in performance and scalability.
3. In order to support ACID developers will have to implement their own code , making their systems more complex.
4. This may reduce the number of safe applications that commit transactions , for example bank systems.
5. NoSQL is not compatible (at all) with SQL.
6. NoSQL are very new compared to relational databases, which means that are far less stable and may have a lot less functionalities.

Classify NoSQL in mongoDB?.

NoSQL database use different data structures compared to relational databases. It makes some operations faster in NoSQL. The suitability of a given NoSQL database depends on the problem it must solve. These databases stores both structured data and unstructured data like audio files, video files, documents etc.

There are four types of NOSQL databases.

They are:-

1. key - value stores.
2. column - oriented
3. Graph - databases.
4. Document oriented databases.

* What is key - value stores?

A. key - value stores are the simplest NOSQL databases. Every single item in a key value database is stored as an attribute name (or "key") together with its value.

Ex:- Riak, Voldemort, and Redis.

key	value
K1	Prakash
K2	Mounya
K3	Tribhuvan
K4	RUDRA
K5	Chaitanya
K6	Ganesh.

* What is column - Oriented?

Column-oriented store columns of data together instead of rows and are optimized for queries over large datasets. Cassandra and HBase are column oriented databases.

Product	
ID	value
1.	Beer
2.	Beer
3.	vodka
4.	Whiskey
5.	Butter scotch
6.	chocolate
7.	Vanilla.

Customer	
ID	customer
1	Chaitanya
2	Tribhuvan
3	Rudra
4	Neera
5	Mourya
6	Manish
7	khan.

* What is Document oriented databases?

A Document databases pair each key with a complex data structure known as a document. Documents can contain many different key-value pair, or key - array pairs, or even nested documents. MongoDB is a Document database.

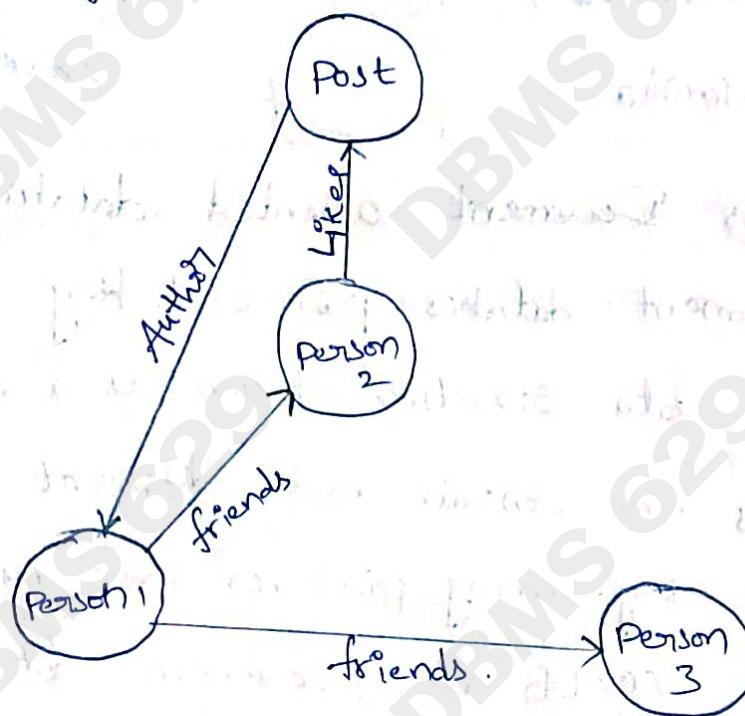
Ex:-

```
{
    name: "sue",
    age : 26,
    status: "A",
    groups : ["news", "sports"] }
```

→ field: value
 → field: value
 → field: value
 → field: value.

Q What is Graph database?

- A. Graph databases are used to store information about networks, such as social connections. Examples include Neo4J and HyperGraph DB. NoSQL databases are rising in popularity as companies apply them to a growing number of use cases.



CHAPTER - 5

BASICS OF MONGODB

Installation of mongoDB:-

Install MongoDB on Windows:-

1. To install MongoDB on windows, make sure you get correct version of mongoDB depending upon your windows Version.
2. To get your windows version, open command prompt and execute the following command.

C:\> wmic os get /osarchitecture

OSArchitecture.

64-bit

c:\>

3. Now extract your downloaded file to c:\ drive or any other location. Make sure the name of the extracted folder is mongodb-win32-i386-[version]. Here [version] of mongodb download.

[version] of mongodb-win32-x86-64-[version]. Here

Version of mongodb download.

4. Next, open the command prompt and run the command.

C:\> mongodb - Win64 - * mongod & / port 27017
move

I dir(s) moved

C:\> cd C:\Program Files\mongodb\bin

5. In case you have extracted the Mongodb at different location, then go to that path by using command cd FOLDER /DIR and now run the above given process. Mongodb requires a data folder to store its files. The default location for the mongodb data directory is c:\data\db. So you need to create this folder using the command prompt. Execute the following command sequence.

c:\>md data

c:\md data\db

6. If you have to install the mongodb at a different location, then you need to specify an alternate path for \data\db by setting the path dbpath is mongod.exe. For the same, issue the following commands. In the command prompt, navigate to the bin directory present in the mongodb installation folder. suppose my installation folder is D:\setup\mongodb\up\mongodb.

c:\users\rajus>d:

D:\>cd "setup"

D:\setup>cd mongodb

D:\setup\mongodb>cd bin

D:\setup\cd mongodb\bin>mongod.exe -dbpath

"d:\setup\mongodb\data".

it will be shown waiting for connection messages on the console output, which indicates that the mongod.exe process is running successfully.

Now to run the mongodb, you need to open another command prompt and use the following command.

```
D:\set up\mongodb\bin>mongo.exe.
```

```
MongoDB shell version : 2.4.6
```

```
Connecting to : test
```

```
>db.test.save({a:1})
```

```
>db.test.find()
```

```
{ "_id": ObjectId("5879b0f65a56a454"), "a": 1 }
```

7. The mongodb installation is successfully completed. If you want to run mongodb, use the following commands.

```
D:\set up\mongodb\bin>mongod.exe --dbpath
```

```
"D:\set up\mongodb\data".
```

```
D:\set up\mongodb\bin>mongo.exe.
```

Differences between RDBMS and MongoDB?

A	RDBMS	MongoDB
	<p>RDBMS is a relational database model in which data is stored in multiple tables.</p> <p>Records are stored as rows in tables, where in table are organized into columns with each column attributed to one data type.</p> <p>It follows a typical schema design comprises of several tables and relationships between them.</p>	<p>MongoDB is an open-source document-oriented database that has no concept of tables, schemas, rows or SQL.</p> <p>MongoDB uses different formats to store data such as document stores, graph, databases, key-value stores, and more.</p> <p>It is based on a schema-less data representation with no regard to the concept of relationship.</p>
	<p>RDBMS databases are vertically scalable meaning when database loads increase, you scale database by increasing the capacity of existing hardware.</p>	<p>MongoDB is a one-size-fits-all database and is considered to be more scalable than the traditional RDBMS database models.</p>

* What are the advantages of MONGODB?

- A 1. Schema less: Mongodb is a document database in which one collection holds different documents.
2. Structure of a single object is clear.
3. No complex joins.
4. Deep query-ability. Mongodb supports dynamic queries on documents using a documents-based query language that's nearly as powerful as SQL.
5. Ease of scale-out MongoDB is easy to scale.
6. conversion / mapping of application objects to database objects not needed.
7. Uses internal memory for storing the working set, enabling faster access of data.

* What are the applications of MongoDB?

1. Large number of objects to store, it splits objects across multiple servers.
2. High write/read throughput and data distribution, Replication lets you scale read and write traffic across multiple servers multiple tenant of data center.
3. Low latency access, Memory mapped storage engine caches documents in RAM, enabling "in memory" operations. Data locality of documents significantly improve latency over joins-based approaches.

4. Variable data in objects, dynamic schema and JSON data model. enable flexible data storage without spare table of complex JSON and provide for an intuitive query language.

5. cloud based deployment.

Installation of Mongo-db in Windows-10

- * Open Chrome and Search www.mongodb.org.
- * Then open a window and Right side of the window we get download option which is latest version.
- * Click on download. Then open the file of mongodb in downloads.
- * Click on mongod file and Right click on that and choose install option.
- * It shows a one pop-up window and click next, next, next, complete, next ----- finish.
- * Open C drive and select the program files then open mongo db afterwards open the server again open the server it shows the folder name which version you installed, then click on bin folder.
- * Copy the path and open Command prompt type this command:-

`> cd` > paste the path here
and click enter.

* Then minimize the command prompt and go to C drive then create new folder and open that and again create a new folder & open

* Again open the command prompt and type command `> mongod` for server connection.

* In desktop search bar we have to search `\environment Variables` → open and edit and system go to environment variables click on edit and click on new now paste the path which we copied before. Then click on ok then ok and close that window.

* And again open another command prompt and type `mongo`.

* Finally we installed mongodb in windows 10.

* How to create a database in mongoDB?

A. MongoDB use DATABASE-NAME is used to create database. The command will create a new database if it doesn't exist, otherwise it will return the existing database.

Syntax:-

use <new db name>

Ex:-

use < peakash >

Dropping of database:-

MongoDB db.dropDatabase() command is used to drop a existing database.

Step 1:- Select the database

Step 2:- drop the database

Syntax:-

use <database-name>

db.dropDatabase()

Ex:-

use employee

db.dropDatabase()

This will be delete the selected database.
If you have not selected any database, then it
will ~~be~~ delete default 'test' database.

Example 1 :-

First, check the list of available databases
by using the command, show dbs.

> show dbs

local 0.78125GB

Mydb 0.23012GB

test 0.23012GB

prakash 0.23412GB

>
If you want to delete newdatabase <mydb>,
then dropdatabase() command would be as
follows.

> use mydb

switched to db mydb

> db. dropdatabase()

> { "dropped": "mydb", "ok": 1 }

>

Now check list of databases.

> show dbs

local 0.78125GB

test 0.23012GB

prakash 0.23412GB

>

List the commands in MongoDB?

- A. 1. Inserting document
- 2. Deleting document
- 3. Updating document
- 4. Sorting document
- 5. limit
- 6. Pretty.
- 7. Query Document.
- 8. Compare
- 9. Drop database.

1. Inserting document:- The insert() method to insert data into MongoDB collection, you need to use MongoDB's insert().

Syntax:-

```
db.<collection-name>.insert (JSON-Document)
```

Ex:-

```
db.prakash.insert ({  
    "names": "Chethan",  
    "gender": "Male",  
    "age": 45  
});
```

2. Deleting document:-

The `remove()` method mongoDB's `remove()` method is used to remove a document from the collection. `remove()` method accepts two parameters. one is deletion criteria. and second is just one flag.

* Deletion Criteria :- (optional) deletion criteria according to documents will be removed.

* Just one :- (optional) if set to true or 1, then removed only one document.

Syntax: `db.<collection-name>.remove(DELETION-CRITERIA, option)`

Ex:-

`db.prakash.remove({ "age": 47 })`

`db.prakash.remove({ "age": 49 }, { 1 })`

3. Updating document:- This command is used for to modify the documents.

Syntax:-

`db.<collection-name>.update(searchCriteria, newdata)`

Ex:-

`db.prakash.update({ "name": "chetan" }, { $set: { "name": "manya" } })`

`db.prakash.update({ "age": 45 }, { $ageSet: { "age": 46 } }, { multi: true })`

Output :-

```
db.prakash.save({  
    "_id": ObjectId("5269834578ad1b26e"),  
    "name": "mowiyah",  
    "gender": "male",  
    "age": 46  
});
```

4. Sorting document:- The `sort()` method helps to sort documents in MongoDB, you need to use `sort()` method. The method accepts a document containing a list of fields along with their sorting order. To specify sorting order 1 and -1 are used. 1 is used for ascending order while -1 is used for descending order.

Syntax:-

```
db.<collection-name>.find(searchCriteria).sort({ "key": 1 })  
db.<collection-name>.find(searchCriteria).sort({ "key": -1 })
```

Example:-
db.prakash.find().sort({ "name": 1 })
ascending order.

```
db.prakash.find().sort({ "age": 1 })  
descending order
```

```
db.prakash.find().sort({ "age": -1 })
```

5. Limit :- This command is used for to show the number of limit documents.

Syntax:-

db.collection-name.find().limit()

Example:-

db.prakash.find().limit(10).

6. pretty:- This command is used for to show in JSON format.

Syntax:-

db.prakash

db.collection-name.find().pretty().

Example:-

db.prakash.find().pretty();

{ "_id": ObjectId("5267834578adib26e"),

 "name": "Mourya",

 "gender": "male",

 "age": 46

}

7. Query Document:- To find() method to query data from MongoDB collection, you can use MongoDB's find() method. In this command we

- have two options. ① to show multiple documents
② to show only one document.

Syntax:-

db.{collection-name}.find(searchCriteria).

db.{collection-name}.findOne(searchCriteria).

Example:-

db.prakash.find({ "gender": "male" }).pretty()

{

 "name": "James",

 "gender": "male",

 "age": 45

,

{

 "name": "Mowry",

 "gender": "male",

 "age": 45

,

{

 "name": "James",

 "gender": "male",

 "age": 45

db.prakash.findOne({ "name": "James" }).pretty()

{

 "name": "James",

 "gender": "male",

 "age": 45

8). Compare:- It is used to compare the elements in the documents. In this command we have two options. One is greater than and another one is less than.

Syntax:-

db. <collection-name>. find ({ \$any item which you want to compare : ({ \$LT : which is less than a number })});

db. <collection-name>. find ({ \$any item which you want to compare : ({ \$GT : which is less greater than a number })});

Example:-

db. prakash. find ({ salary : ({ \$LT : 10000 })});

db. prakash. find ({ salary : ({ \$GT : 20000 })});

9. Drop database:- This command is used for dropping the database. This is only when which data base is opened, that only we can drop the data base.

Syntax:-

db. run command ({ drop database : 1 });

Example:-

db. runcommand ({ drop database : 1 });

O/P

Local 0.0000,

Admin 0.0000

Show dbs →

config .00.000

Mongodb commands:

1. Use <db name>
2. db.createCollection("collection_name");
3. db.collection_name.drop();
4. Show dbs
5. show collections
6. db.collection_name.insert({ "values": "3"});
7. db.collection_name.update({ old data }, { new data }, { \$set, ins, dec });
8. db.collection_name.find({ particular one });
9. db.collection_name.find().count();
10. db.collection_name.find({ any item in documents we want to compare : { \$LT : maximum value } });
11. db.collection_name.find({ any item in documents we want to compare : { \$GT : minimum value } });
12. db.collection_name.find().sort({ anyone : Value });
13. db.collection_name.find().sort({ Anyitem : -Value });
14. db.employee.find().limit(values that you want);
15. db.collectionname.find().pretty();
16. db.collection_name.insertMany([{ values }]);
17. db.collection_name.deleteOne({ "which want to delete" : "value" });
18. db.collection_name.deleteMany("multiple documents" : "values");

19. db.gruncommand ({dropdatabase : 1});
20. db.'name'.drop();

SQL COMMANDS

Numeric Functions:-

1. ABS(num) : Returns Absolute value of given number.

Ex: SQL> SELECT ABS(-10) FROM DUAL;

ANSWER :-
10

2. CEIL (num) : Return the smallest integer greater than given number.

Ex:-

SQL> SELECT CEIL(12.456) FROM DUAL;

ANSWER :-
12.5

3. FLOOR(num) : Returns the largest integer smaller than given number than the given value.

Ex:-

SQL> SELECT FLOOR(123.456) FROM DUAL;

ANSWER :-
123

4. LN(num) : It returns natural logarithm value of "num".

Ex:- SQL> SELECT LN(10) FROM DUAL;

ANSWER :-
2.3025809

5. $\text{LOG}(m,n)$: It returns logarithm of "n" with base "m".

Ex:- `SELECT LOG(100,10) FROM DUAL;`

$\text{LOG}(100,10)$

5

6. $\text{MOD}(m,n)$: It returns remainder of "m" divided by "n".

Ex:-

`SQL> SELECT MOD(10,3) FROM DUAL;`

$\text{MOD}(10,3)$

1

7. $\text{POWER}(m,n)$: It returns value equal to "m" raised by "n".

Ex:- `SQL> SELECT POWER(10,2) FROM DUAL;`

$\text{POWER}(10,2)$

100.

8. $\text{ROUND}(m,n)$: It round the given float-valued number "m" to the "n" places after the decimal.

Ex1:-

`SQL> SELECT ROUND(1.23456) FROM DUAL;`

$\text{ROUND}(1.23456)$

1.

Ex2:- `SQL> SELECT ROUND(1.23456,3) FROM DUAL;`

$\text{ROUND}(1.23456,3)$

1.235.

9. SQRT(m): It calculates square root value
of number "m".

Ex:- SELECT SQRT(9) FROM DUAL;

SQL > SQRT(9)

10. TRUNC(m,n): It truncates given float-valued number "m" to "n" places after the decimal.

Ex:-
SQL > SELECT TRUNC(1.23456,3) FROM DUAL;

TRUNC(1.23456,3)
— — — — —
1.234

Ex:-
SQL > SELECT TRUNC(1.23456) FROM DUAL;

TRUNC(1.23456)

— — — — — : Comma in DUAL

CHARACTER FUNCTIONS

11. UPPER(str): It converts all letters in the given string "str" in to uppercase.

Ex :- SQL > SELECT UPPER('abcDEFgh')

FROM DUAL;

UPPER

— — — —

ABCDEF GH

12. LOWER(str): It converts all the letters in the given string "str" in to Lower Case.

Ex:- SQL > SELECT LOWER('ABCDEfg') FROM DUAL;

LOWER

— — — —

abcdefg

13 INITCAP (str) : It converts first letter is capital letter and all are in lowercase

Ex:-

SQL > SELECT INITCAP ('ABCDEF') FROM DUAL;

INITCAP

Abcdef

14 LENGTH (str) : It returns the number of characters in the given string (including spaces).

Ex:-

SQL > SELECT LENGTH ('ABCD') FROM DUAL;

LENGTH('ABCD')

4.

15 SUBSTR (str,m,n) : will extract "n" characters from the given string starting from " m^{th} " position.

Ex:-

SQL > SELECT SUBSTR ('ABCDE FG', 2,3) FROM DUAL

SUB

BCD

16. INSTR (string,str,m,n) : It displays n^{th} occurrence of "str" in the string "string" starting from "m".

Ex:-

SQL > SELECT INSTR ('Trying to keep the things as sample as possible','AS',1,2) From DUAL

INSTR

37.

17. LPAD():- This function is used to left pad the given string with specified character or string.

Ex:-

SQL > SELECT LPAD('BCD', 4, 'A') FROM DUAL;

LPAD

- -

ABCD.

18. RPAD():- This function is used to left pad the given string with specified character or string.

Ex:-

SQL > SELECT RPAD('BILL', 12, 'CLINTON') FROM DUAL;

RPAD ('BILL',

- - - - -

BILL CLINTON

19. LTRIM():- This function removes specified string from the given string if it is there to the left of given string.

Ex:-

SQL > SELECT LTRIM('GEORGE BUSH', 'GEORGE')

FROM DUAL;

- - -

BUSH

20. RTRIM():- This function removes specified string from the given string if it is there to the right of given string.

Ex:-

SQL > SELECT RTRIM('TONY BLAIR', 'AIR') FROM DUAL;

RTRIM ()

- - - - -

TONY BL

21 ASCII() :- Displays equivalent ASCII value of a character.

Ex:-

SQL > SELECT ASCII('A') FROM DUAL;

ASCII('A')

65.

22 ADD_MONTHS(Date, n) : Adds n months to be specified date.

Ex:-

SQL > SELECT ADD_MONTHS('1-JAN-05', 5)
FROM DUAL;

ADD_MONTHS

01-JUN-05

23 LAST_DAY(date) : Gives last date of the specified month (date)

Ex:-

SQL > SELECT LAST_DAY('1-JAN-05') FROM DUAL;

LAST_DAY (

31-JAN-05

24. MONTHS_BETWEEN (date 1, date 2) : It gives difference between the two dates date 1, date 2 in months.

Ex:-

SQL > SELECT MONTHS_BETWEEN ('31-DEC-05', '1-JAN-05')

FROM DUAL;

MONTHS_BETWEEN ('31-DEC-05', '1-JUL-05')

11.9677419.

5. NEXT_DAY (date, day) : It gives date of the next occurrence of the specified day after the given date.

Ex:-
SQL> SELECT NEXT_DAY('01-JAN-05', 'FRI')
FROM DUAL;

Next_DAY (' Next Friday after 1-Jan-05 is on 7-jan-05)

07-JAN-05.

26.

SYSDATE Functions :-

USER

Ex:-
SQL> SELECT USER FROM DUAL;

USER

SCOTT

SYSDATE function displays system date

Ex:-
SQL> SELECT SYSDATE FROM DUAL;

SYSDATE

01-JAN-05

01-JAN-05

CONVERSION FUNCTIONS

8. TO_CHAR : This function is used to convert a date or number to character string.

Ex:-

SQL> SELECT TO_CHAR(SYSDATE, 'DAY DD MONTH YYYY')
FROM DUAL;

TO_CHAR (SYSDATE, 'DAY DD MONTH

SATURDAY 01 JANUARY 2021

29. TO_DATE (string): This function converts a string into an oracle date.

Ex:-

SQL> SELECT TO_DATE ('01 JANUARY 2021', 'DD MONTH
4444') FROM DUAL;

TO_DATE('01 JANUARY 2021', 'DD MONTH

01 - JAN - 05

30 NVL() Function: This function is used to substitute ^{null} any value with a user-defined value.

Consider the following data from EMP table of

SCOTT,

Ex:-

SQL> SELECT EMPNO, ENAME, SAL, COMM FROM EMP;

EMPNO	ENAME	SAL	COMM
100 1	John	800	
100 2	Bumrah	1600	300
100 3	Smith	1250	
100 4	Kohli	2975	500
100 5	Dhawan	1270	
100 6	Ashwin	1970	1400
100 7	Dhoni	32500	
100 9	Rahul	1300	
101 1	Pant	1600	0
101 2	Chahal	1700	

In the above table except for 1002, 1004, 1006 and 1011 all other others commissions are null.

To display their commission as "0" (zero)

We can use NVL() function as shown below.

SQL> SELECT EMPNO, ENAME, SAL, NVL(COMM, 100)

FROM EMP;

EMPNO	ENAME	SAL	NVL(COMM, 100)
1001	John	800	100
1002	Bumrah	1600	300
1003	Smith	1250	500
1004	Kohli	2975	100
1005	Dhawan	1270	100
1006	Ashwin	1970	1200
1007	Dhoni	32500	100
1008	Rahul	1300	100
1009	Pant	1600	0
1010	Chahal	1700	100
1011			
1012			

31. DECODE () function:-

Used for substituting values in the query result.

~~selected empno, decode~~

SELECT bank_name,

DECODE (Bank_id, 001, 'SBI',

002, 'ICICI',

003, 'CANARA',

'Gateway') result

FROM banks;

AGGREGATE FUNCTIONS

32. MAX (): A function used to find maximum value among the given values.

Ex:- SQL > SELECT MAX(SAL) FROM EMP;

MAX (SAL)
5000

33. MIN (): It returns the minimum of the values in a column.

Ex:-

SQL > SELECT MIN(HIRE DATE) FROM EMP;

MIN (HIRE DATE)
17 - DEC - 1999.

34. AVG (): Gives the average of values in a column.

Ex:-

SQL > SELECT AVG(SAL) FROM EMP;

AVG (SAL)
2073.21429.

35. SUM (): Sum of the values in a column.

Ex:-

SQL > SELECT SUM(SAL) FROM EMP;

SUM (SAL)
29056272.

36. COUNT(<columnname>):

Gives no. of not null ENTIRE (VALUES) in that column.

Ex:-

SQL > select COUNT(*) FROM DUAL;

COUNT (*)

1

Ex2:-

SQL > SELECT COUNT(COMM) FROM EMP'S

COUNT (COMM)

4

37. COUNT (*): Gives number of columns in a table.

Ex:-

SQL > SELECT COUNT(*) FROM EMP;

COUNT (*)

14

38. AVG (COLUMN NAME):-

The function returns the average of columns value.

Ex:-

List the average salary of employees working in department 30.

SELECT avg(sal) from emp where deptno=30.

CREATE:

CREATE Table Table_name

(column-name1 datatype,

column-name2 datatype,

:

:

column-nameN datatype

);

Greatest of three Numbers:

Declare

a number := 46;

b number := 67;

c number := 21;

Begin

If a > b

AND a > c Then

dbms_output.put_line ('Greatest number is ' || a);

Elseif b > a

AND b > c then

dbms_output.put_line ('Greatest number is ' || b);

Else

dbms_output.put_line ('Greatest number is ' || c);

End if;

END;