DATABASE MANAGEMENT SYSTEM

DATA: The raw fact which is given or fed into computer for processing is termed as data.
INFORMATION: A set of inter-related data is known as Information.

DATABASE: Collection of inter-related information is known as database.

DATABASE MODELING: It is a way to represent / store database within a system.
Types of data modeling are:

1) Hierarchical Model
2) Network Model
3) Relational Model

RELATIONAL MODEL: The relational model was first proposed by E.F .Codd in his seminal paper 'A relational model of data for large shared data banks' in 1970. Due to the simplicity of relational data model and its strong foundation in mathematics , it attracted database researchers around the world.

TABLE - STUDENT

Roll	Name	Course	Add.
101	Ranjan	MCA	Patna
102	Sanjeev	BCA	Ranchi
103	Pallavi	MCA	Patna

Relation : A relation is table with columns and rows.

Attribute : An attribute is a named column of a relation . Relations are created to hold information about the objects to be represented by database. Columns in a relation are called attribute. For Example- roll no. , name, course etc.

Domain : A domain for an attribute is a set of allowable values that can be stored for that attribute. For example- Roll No. attribute for student relation can have only Numeric values greater than Zero.

Tuple : A tuple is row of a relation.
OR The elements of a relation are the rows or tuples in relation.
Candidate key : A candidate key (sometimes also called simply a key) is a superkey such that no proper subset of it uniquely identifies rows of a relation.

Primary Key : The candidate key that selected to identify tuples uniquely in a relation is called primary key. A relation can have only one primary key. For example –Roll no.

Alternate Key : The candidate keys that are not selected as the primary key of a relation are called alternate. In example Student table – name , phone etc.

Foreign Key : Foreign key is an attribute or a set of attributes in relation that matches the candidate key of some other relation.

RELATIONAL ALGEBRA :

Basic operations:

– ***Selection*** **(σ)** Selects a subset of rows from relation.

– ***Projection*** **(π)** Deletes unwanted columns from relation.

– ***Cartesian-product*** **(×)** Allows us to combine two relations.

– ***Union*** **(U)** Tuples in reln. 1 and in reln. 2.

UNION: consider two relations R and S . The UNION of relations R and S is again a relation that contains all the tuples of R and S or tuples which are both R and S , Duplicate Tuples being eliminated. It is denoted by R U S.

EMPLOYEE

CODE	Name
101	Raman
102	Harmanjeet
103	Satish
201	Veenu
203	Himani

MANAGER

CODE	Name
201	Veenu
202	Kushal pal
203	Himani

EMPLOYEE U MANAGER

Code	Name
101	Raman
102	Harmanjeet
103	Satish
201	Veenu
202	Kushal pal
203	Himani

CARTESIAN PRODUCT :The Cartesian product of two relations R and S is the relation which is

concatenation of every tuple of relation R with every tuple of relation S.

TOY

Toy Name	Price
Teddy bear	300
Ping Pong	400

COLOUR

Colour
Red
Green

CARTESIAN PRODUCT OF TOY x COLOUR

TOY x COLOUR

Toy Name	Price	Colour
Teddy bear	300	Red
Teddy bear	300	Green
Ping Pong	400	Red
Ping Pong	400	Green

SELECTION : The SELECTION operation is used to select a subset of tuples from a relation that satisfy a particular condition called selection condition. SELECTION operation can also be understood as a filter that keeps only those tuples which satisfy the mentioned condition.
In general the SELECTION operation is denoted by …σ…

σ <Selection Condition> $^{(R)}$

Consider following EMPLOYEE relation :

EMPLOYEE

Emp_No	EName	Job	Salary	Dept_No
101	Raman	President	50000	10
102	Ranjan	Manager	12000	30
103	Satish	Salesman	9000	30
201	Veenu	Analyst	14000	20
203	Himani	Clerk	9000	20
204	Kushal pal	Salesman	10000	30

SELECTION operation is : σ Dept_No=30 **(EMPLOYEE)**

Results of above operation is :

Emp_No	EName	Job	Salary	Dept_No
102	Ranjan	Manager	12000	30
103	Satish	Salesman	9000	30
204	Kushal pal	Salesman	10000	30

PROJECTION : The PROJECTION operation selects certain columns from the table and discards other columns, PROJECTION works on a single relation R and outputs a vertical subset of R , extracting the value of specified attributes .

In general the PROJECTION operation is denoted by

π <Attribute list> (R)

For example – π EName, Job, Salary (EMPOLYEE)

Results of above operation is :

EName	Job	Salary
Raman	President	50000
Ranjan	Manager	12000
Satish	Salesman	9000
Veenu	Analyst	14000
Himani	Clerk	9000
Kushal pal	Salesman	10000

What is Normalization?

Normalization is the process of efficiently organizing data in a database.

There are two goals of the normalization process:

1.Eliminating redundant data

2.Ensuring data dependencies make sense .

First Normal Form (1NF)

First normal form (1NF) sets the very basic rules for an organized database:

Eliminate duplicative columns from the same table.

Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

Second Normal Form (2NF)

Second normal form (2NF) further addresses the concept of removing duplicative data:

Meet all the requirements of the first normal form.

Remove subsets of data that apply to multiple rows of a table and place them in separate tables.

Create relationships between these new tables and their predecessors through the use of foreign keys.

Third Normal Form (3NF)

Third normal form (3NF) goes one large step further:
Meet all the requirements of the second normal form.
Remove columns that are not dependent upon the primary key.

Fourth Normal Form (4NF)

Finally, fourth normal form (4NF) has one additional requirement:
Meet all the requirements of the third normal form.
A relation is in 4NF if it has no multi-valued dependencies.

HISTORY OF SQL : In **1974** D. Chamberlin who was also from IBM's San Jose Laboratory , Defined a Language called Structured English Query Language or **SEQUEL** . A revised version of SEQUEL called SEQUEL/2 was defined in **1976**.This name was subsequently changed to **SQL.** In **1986** a standard for sql was defined by **ANSI** (American National Standards institute). In **1987** SQL was adopted for use with RDBMS as an international standard by **ISO** (International organization for standardization).
SQL Commands fall into three categories –

- **DDL** (Data Definition Language)
- **DML** (Data Manipulation Language)
- **DCL** (Data Control Language)

Data Definition Language (DDL) : DDL is used to create and remove database objects.
Example : CREATE , ALTER , DROP
Data Manipulation Language (DML) : DML is used to manipulate the data in the database.
Example : SELECT , INSERT , UPDATE , DELETE
Data Control Language (DCL) : DCL is used to control the kind of data access to the database, and also for Transaction control.
Example:
GRANT and REVOKE are used for Data Access Control
COMMIT and ROLLBACK , SAVEPOINT are Transaction Control Commands.

SQL DATA TYPES:

- **NUMBER** : A Number data type is a standard data type. It may contain a number with or without a decimal point and sign.
- **CHARACTER** : CHARACTER data type is a standard data type that contain fixed length character data.
- **DATE** : DATE is a standard data type which is used to stored date and time values ranging from January 1 , 4712 B.C to December 31, 9999. it is defined simply as DATE.

SQL COMMAND :

- **CREATE COMMAND** : It is a type DDL statement which used to create a table in exiting database.

SYNTAX :

```
CREATE TABLE <table name>
(
Field-name   Field-type (width),
Field-name   Field-type (width),
Field-name   Field-type (width)
);
```

```
SQL>CREATE TABLE Student
 (
Roll Number (10),
Name char(25),
Birth     Date
);
```

ALTER COMMAND : It is a type of DDL statement that is used:

(a) To ADD a new column.

(b) To Modify an existing column by modifies its sizes.

(c) To drop a constraint of an existing field.

SYNTAX :

- ALTER TABLE <Table name>
  ADD Field-name Field –type (width);
- ALTER TABLE <Table name>
  MODIFY Column-name Type (width);

SQL> ALTER TABLE Student ADD Phone Number(12);

SQL> ALTER TABLE Student MODIFY Name char(30);

SQL> ALTER TABLE Student ADD PRIMARY KEY(Roll_No);

SQL>ALTER TABLE Student DROP PRIMARY KEY ;

DROP COMMAND: DROP statement used for to delete a predefined structure within a database.

SYNTAX:

DROP TABLE <Table-name>;

SQL> DROP TABLE Student;

INSERT COMMAND: It is an DML statement tool that is used to insert new records in the table.

SYNTAX:

INSERT INTO <table-name> Values(V1 , V2 , V3 , …….);

SQL> INSERT INTO Student values (102,'Ranjan','01-Feb-1982');

SQL> INSERT INTO Student values (103,'Sanjeev','01-Dec-1986');
SQL> INSERT INTO Student values (104,'Abhishek','06-Mar-1987');

SQL> INSERT INTO Student values(&Roll_No,'&Name','&Birth');
SQL> Enter Roll_No : 106
SQL> Enter Name : Sanjay
SQL> Enter Birth : 12-apr-1990
SQL> /

NOTE:

1. Character values are typed within single quotes.
2. Date values are typed within single quotes.
3. Numeric values are typed as they are.

SELECT COMMAND: It is a DML Tool .It is used to different types of query in a database.

SYNTAX:

SELECT <list name> / <where condition> / * FROM <Table name> ;

To list all the fields from table Student.
SQL> SELECT * FROM Student;
SQL>SELECT Roll_No, Birth FROM Student;

SQL> SELECT * FROM Student WHERE Roll_No=103;

SQL>SELECT * FROM Student WHERE Name LIKE 'R%';

UPDATE COMMAND: it is an DML tools which is used to update field label data on an inserted records.

SYNTAX:

UPDATE <table-name> SET <Field-name> = value WHERE Field=value;

SQL> UPDATE Student SET Phone=222134 WHERE Roll_No=102;

SQL>UPDATE Student SET Name=’Prakash Ranjan’ WHERE Roll_No=103;

SQL> UPDATE Student SET Name=’Rakesh Kumar’ , Birth=’13-Mar-1990’ WHERE

Roll_No=104;

DELETE COMMAND: It is also DML tool .it is used to delete a record or multiple records from a table.

SYNTAX:

DELETE FROM <table-name>;

DELETE FROM <table-name>WHERE <condition>;

SQL> DELETE FROM Student;

(Will Delete all rows from Student table.)

SQL> DELETE FROM Student WHERE Roll_No=103;

(will delete only those rows which contain 103 in the Roll_No Column).

TABLE – Teacher

TCode	Name	Age	Dept	DateofJoin	Salary	Sex	City
1	Ranjan	24	Computer	25-Jan-2005	24000	M	Pat
2	Sandeep	30	Physics	24-Mar-2001	14000	M	Sbg
3	Sharmila	34	Chemistry	12-Dec-1998	20000	F	Pat
4	Sanjeet	26	Physics	01-Jul-1980	23000	M	Del
5	Sangeeta	25	Chemistry	12-Mar-1987	35000	F	Del
6	Rakesh	35	Physics	17-Feb-1981	17000	M	Sbg
7	Pallavi	25	Computer	16-Mar-2005	18000	F	Pat

DISTINCT:

=> (To Select the single occurrence of any value from table this command is used)

SQL> SELECT DISTINCT Dept FROM Teacher;

(There may be multiple records for computer , physics and chemistry and above command will display a single record for computer , physics and chemistry dept)

OUTPUT:

DEPT
Computer
Physics
Chemistry

BETWEEN n1 AND n2

This operator can be used to retrieve rows which fall within a range.

SQL> SELECT * FROM Teacher WHERE Salary BETWEEN 15000 AND 20000;

Output:

TCODE	NAME	AGE	DEPT	DATEOFJOIN	SALARY	SEX	CITY
6	Rakesh	35	Physics	17-FEB-81	17000	M	Sbg
3	Sharmila	34	Chemistry	12-DEC-98	20000	F	Pat
7	Pallavi	25	Computer	16-MAR-05	18000	F	Pat

IN : The IN operator is used to specify a list of possible values for column.

SQL> SELECT * FROM Teacher WHERE City IN ('Pat' , 'Sbg');

(Will retrieve rows where the city column has data which is equal to Pat or Sbg)

Output:

TCODE	NAME	AGE	DEPT	DATEOFJOIN	SALARY	SEX	CITY
2	Sandeep	30	Physics	24-MAR-01	14000	M	Sbg
6	Rakesh	35	Physics	17-FEB-81	17000	M	Sbg
1	Ranjan	24	Computer	25-JAN-05	24000	M	Pat
3	Sharmila	34	Chemistry	12-DEC-98	20000	F	Pat
7	Pallavi	25	Computer	16-MAR-05	18000	F	Pat

This retrieve can also be coded as :

SQL> SELECT * FROM Teacher WHERE City='Pat' OR City='Sbg';

GROUP BY Clause : At times we need to divide the rows in a table into smaller groups of information. This can be done by using GROUP By clause.

```
SQL> SELECT Dept , COUNT(TCode), SUM(Salary)
        FROM  Teacher GROUP BY Dept ;
```

Output:

DEPT	COUNT(TCODE)	SUM(SALARY)
Computer	2	42000
Physics	3	54000
Chemistry	2	55000

```
SQL> SELECT Dept, COUNT(TCode), SUM(Salary)
      FROM Teacher
      WHERE Sex='M'
      GROUP BY Dept;
```

DEPT	COUNT(TCODE)	SUM(SALARY)
Computer	1	24000
Physics	3	54000

HAVING Clause : HAVING clause is used to restrict rows resulting after the application of GROUP BY clause .

- Rows are grouped according to the columns in GROUP BY clause.
- Group function is applied.
- Groups matching the HAVING clause are displayed.

```
SQL> SELECT Dept, COUNT (TCode), SUM(Salary)
     FROM Teacher
     WHERE Sex='M'
     GROUP BY Dept
     HAVING   SUM(Salary)>50000;
```

Output:

DEPT	COUNT(TCODE)	SUM(SALARY)
Physics	3	54000

Dept	COUNT(TCode)	SUM(Salry)

ORDER BY Clause : The order of rows returned in query is same as the order in which rows were inserted into the table. ORDER By clause is used to sort the rows returned as a result of SELECT statement. The data in the table is not sorted; only the results shown on the screen are sorted . ORDER BY Clause is always specified as last clause in SELECT statement. Keywords used with ORDER BY Clause are:

ASC : For ascending order(It is the default)

DESC : For descending order.

SQL> SELECT Name , Age ,Salary FROM Teacher
WHERE Sex=’F’ ORDER BY Salary;

Output:

NAME	AGE	SALARY
Pallavi	25	18000
Sharmila	34	20000
Sangeeta	25	35000

SQL> SELECT * FROM Teacher
ORDER BY Date of join DESC ;

Output Results:

TCode	Name	Age	Dept	Date of join	Salary	Sex	City

SQL FUNCTION :

- **SUM** : Finds Sum Of Numeric values.
- **AVG** : Finds average of numeric values.
- **COUNT** : Counts the number of rows in a table.
- **MAX**: Finds Maximum among all the values in the column.
- **MIN**: Finds minimum among all the values in the a column.

SQL> SELECT SUM(Salary) FROM Teacher ;

SQL> SELECT AVG(Salary) FROM Teacher ;

SQL> SELECT COUNT(TCode) FROM Teacher ;

SQL> SELECT MAX (Salary) FROM Teacher ;

SQL> SELECT MIN (Salary) FROM Teacher ;

****

THE END

BY-PRAKASH RANJAN(MCA)
SAHIBGANJ COLLEGE , SAHIBGANJ
CONTACT :9905110753 &9955501231