**DBMS**:

===================================================================

DBMS is a technique to store and manipulate the data for future use;

In different type of application ,we need to persist the data which can be used in future,DBMS is a software which can store,manage and manipulate the data.

**RDBMS:**

It stands for relational database management system.RDBMS is the collection of inter related tables where each table may be multiple records which is logically related.

**Table:**

Tables are the collection of logically related data.many RDBMS software/flavours are available in market,you can any one with your application that is suitable:

Ex: MYSQL ,MSSQL , ORACLE , MSACCESS………….

**SQL:**

SQL stands for structure query language.SQL provides some pre defined command that is used perform different operation in RDBMS.

By using SQL command we can create the data,fetch/retrieve the data ,manipulate the data and secure the data.

**SQL:**

**===================================================================**

**A.DDL:**

**……………………………………………………………………………………………**

data definition language.->apply on the schema of the table.

**EX:**

**1.Create->**table definition/schema..

**2.Aalter->**add a column/pre-existing column delete/pre-existing name/data/manipulate

**3.Truncate->**reset the schema

**4.Drop->**remove the table

**B.DML:**

**……………………………………………………………………………………………………………………..**

data manipulation program. ->Record apply(create ,manipulate,fetch).

**Ex:**

**1.Insert->**Add a new record. it is used to add a new record/row in a table

**Syntax:**

**Insert** into table\_name value(value1,value2…………valuen);

**Syntax to add data into specific columns:**

Insert into table\_name(column\_name\_1,column\_name\_2,…….,column\_name2…

….column\_name\_n)values(value1,value2………….valuen)

**Add bulk record via insert:**

Insert into table\_name values

(value\_1……………………….value\_n),

(value\_1………………………value\_n);

**2.Select->** fetch/retrive any part of data->

**select used to return any part of data from table.**

select used to retrive data in different way from the database.select is mostly used query of sql.

The only purpose to save the data is that we may used stored data in future,ifyou want to search any part of data from the complex collection….select used to fetch data in different manner.

**Syntax:**

Select column\_nmes(s)from table\_name;

Example:

Select column\_name\_1, column\_name\_1 from table\_name

Note:without condition select returns all present row in the table.

**Select without condition (where clause):**

Select column\_name(s)from table\_name where<condition>

**Operator used in condition:**

= (equals to)

>(greater than)

<(less than)

>=(greater than or equals to)

<=( greater than or equals to)

<>(not equals to) ->day1:1,2,3,4,2.2,2.1

In, not in, between, null, is null, like

……………………………………………………………………………………………………………………………….

Querry used 07-08-2024:

……………………………………………………………………………………………………………………………….

create table tbl1\_student(

rollno int primary key,

name varchar(50)not null,

address text default'Lucknow' ,

mobno bigint not null,

fee float check (fee>=0),

dob date

);

insert into tbl1\_student

values

(1,'techpile','gkp',23458732,40000,'2012-04-24')

insert into tbl1\_student

values

(2,'sudha','lucknow',23578346,50000,'2024-08-07')

insert into tbl1\_student(rollno ,name,mobno,fee)

values(3,'ramesh',37548365,200)

insert into tbl1\_student(rollno ,name,mobno,fee)

values

(4,'suresh',754683954,5000),

(5,'suresh',754683954,5000)

select rollno,name,mobno,fee from tbl1\_student

select \*from tbl1\_student

select \*,name,rollno from tbl1\_student

select name,rollno+10 from tbl1\_student

select name,rollno+10 as id from tbl1\_student

select name,rollno+10,fee,fee/10 as exam\_fee from tbl1\_student

select name,rollno+10,fee,fee/10 as exam\_fee,\* from tbl1\_student

**3.update->** change any cell/value of any row.

Note: without condition,column of every row will be updated.

UPDATE,SELECT,DELETE-WITH CONDITION

UPDATE TAL\_STUDENT1 SET NAME=’SHIVAM’,DOB=’2002-07-02’ WHERE DOB IS NULL

**4.delete->** remove a row FROM THE TABLE. Delete does not remove record from the table permanently, deleted records can be rollbacked.

Syntax:

Delete from table\_name

Delete from table\_name where <condition>

…………………………………………………………………………………………………………………………………………………….

JOINING:

TBL\_STUDENT:-parent table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ROLLNO | NAME | EMAIL | PASSWORD | FEE |
| 1 | RAM | dhgtihyj@gmail.com | 1 | 243546547 |
| 2 | shyam | dkjfrlkgh@gmail.com | 2 | 5436547 |

Primary key=unique +not null,unique

Tbl\_address:-child table

|  |  |  |  |
| --- | --- | --- | --- |
| Address | city | pincode | Id-fk |
| Lko | Lko | 235453 |  |
| gkp | gkp | 235346 |  |

Tbl-fee-child:-child

AUTOINCREMENT:

ID INT PRIMARY KEY IDENTITY(SEED,INCREMENT)

ID INT PRIMARY KEY IDENTITY(100,1),

USERID SMALLINT FOREIGN KEY REFRENCES TBL\_B1 (ROLLNO)

CREATE TABLE TBL\_ADDRESS

(

ID INT PRIMARY KEY IDENTITY(100,1),

ADDRESS TEXT,

CITY VARCHAR(100),

PINCODE INT,

USERID SMALLINT FOREIGN KEY REFERENCES TBL\_B1 (ROLLNO)

)

INSERT INTO TBL\_ADDRESS VALUES('PLOT NO 43','LUCKNOW',272162,1),

('PLOT NO 44','LUCKNOW',272162,1)

**Foreign key:** It is a column of table which value is inherited by primarykey column of another table.A foreign key type column may have duplicate values but may have only those value that is already present in parent table

**C.DCL:**

**……………………………………………………………………………………………….**

data control language.-> to create users.and grant permission,on the controlling of users of DB;

**Ex:grant,revoke,rename;**

**D.TCL:**

**……………………………………………………………………………………………………………………….**

transaction control language.-> use to manage the consistency , and transaction.

**Ex:save transaction/begin transaction/savepoint,commit,rollback;**

===================================================================

**WEB API: A**pplication programming interface.

**Create :**create command is used to create a new database object like table ,stored procedure,index,view,cursor,database etc.

**Syntax:**

Create database database\_name;

**Syntax to create a table:**

Create table<table\_name>

(

Column\_name dta\_type(size),

Column\_name data\_type,

.

.

Column\_name data\_type(size)

);

**Data type:**

**Whole number:** smallint,int,bigint,

**Decimal numbers:** float,money

**String value:**char(0-255),varchar(0-255),nvarchar(0-255),text;

**(Char()->**provides fixed memory location in a column.

**Varchar()->**provides variable size of memory location to a column.

**Nvarchar-()>**multi-language string value.

**Text->**large sized of string value)

Create table student(

Address char(100)

);

**Data type value:**

Date-yyyy-mm-dd,datetime yyyy-mm-dd hh:mm:ss am/pm

**Boolean value:**bit

Small,int,bigint,float,date,datetime,

-------------------------------------------------------------------------------------------------------------

**Constraints:**

Constraints are use to apply some extra rules regulation on the column definition.

1.**not null:** it restrict null values to be inserted in a column.

2.**default:** default constraints adds a default value to be inserted at the place of null.

3.**check**: it check a given condition before adding any value in a column

4.**primary key** : a primary key type column does not accept null value not duplicate value.it is a normal assumption that every tablr should have atleast one primary key type column so that we can identify every row uniquely by using value of primary key column.

5.foreign key

6.**unique key:** it is same as primary key which is used to identify every row uniquely in the table .But it can hold a single NULL value.

Unique key=primary key+null key

Create table tbl\_student1

(

Roll\_no smallint primary key,

Name varchar(50) not null,

Address text default’ Lucknow’,

Mobile\_no bigint not null,

Fee float check fee>=0

);

Server->multiple database->

CREATE TABLE TBL\_DEPARTMENT(

SR INT PRIMARY KEY IDENTITY(1,1),

NAME VARCHAR(100)NOT NULL,

SEAT INT,

HOD VARCHAR(100)

);

INSERT INTO TBL\_DEPARTMENT VALUES

('COMPUTER SCIENCE',60,'MR.RAM'),

('INFORMATION TECHNOLOGY',80,'MR.SHIVAM'),

('ELECTRONICS ENGINEERING',100,'MR.SHIV')

SELECT\*FROM TBL\_DEPARTMENT

CREATE TABLE TBL\_STUDENT(

NAME VARCHAR (100)NOT NULL,

EMAILID VARCHAR (100) PRIMARY KEY,

PASSWORD VARCHAR(100),

DEPARTMENT INT FOREIGN KEY REFERENCES TBL\_DEPARTMENT(SR)

);

SELECT\* FROM TBL\_STUDENT

ALTER TABLE TBL\_STUDENT ADD T\_FEE INT;

INSERT INTO TBL\_STUDENT VALUES

('Ritik','r@gmail.com','23479067',1,40000),

('Meera','meera@gmail.com','23471417',2,60000),

('Siya','siya@gmail.com','2347976',3,30000)

//SELECT STUDENT NAME,DEPARTMENT NAME,AND HOD NAME USING BOTH TABLE

//TBL\_STUDENT - DEPARTMENT :TBL\_DEPARTMENT (SR)

JOINING: Joining is used to connect record of two or more tables.

By using joining we can select records from more than one table in a virtual table but all tables should be connected with each other via primary key & foreign key relation.

If two tables has foreign key relationship, we can apply joining on that.

Joining is used to map record of one table to records of another table based on a common column.

In SQL joining are mainly 4 types:

1. Inner Joining:

Inner joining maps records of first table to the record of second table only when both table has common value.

Syntax:

Select table\_name.column\_name,

Table\_name.column\_name from first\_table\_name

Inner join second\_table\_name on first\_table\_name.common.column=second\_table\_name.common\_column\_name

Select t1.c1,t2.c2from t1 join t1 on t1.primary\_key=foreign\_key

EXAMPLE:

SELECT TBL\_STUDENT.NAME,TBL\_DEPARTMENT.NAME FROM TBL\_DEPARTMENT

INNER JOIN TBL\_STUDENT ON TBL\_STUDENT.DEPARTMENT=TBL\_DEPARTMENT.SR

SELECT TBL\_STUDENT.NAME,EMAILID,PASSWORD,TBL\_DEPARTMENT.NAME FROM TBL\_STUDENT

INNER JOIN TBL\_DEPARTMENT ON TBL\_STUDENT.DEPARTMENT=TBL\_DEPARTMENT.SR

SELECT \* FROM TBL\_STUDENT LEFT JOIN TBL\_DEPARTMENT

ON TBL\_STUDENT.DEPARTMENT=TBL\_DEPARTMENT.SR

Note: Inner join returns only common records from both tables.

1. Left joining
2. Right joining
3. Full joining

**AGGREGATE FUNCTION**: Aggregate function are used with select command .It isused to perform mathematical task on the values and returns a single output .

**1.SUM (column\_name)**: add all the values of the given column and returns the output.

**2.IN (column\_name)**: returns smallest values present in the column.

**3.MAX ( column\_name)**: returns largest value present in the column.

**4.AVG (column\_name)**: calculate average of values of the given column and returns the output.

**5.COUNT(column\_name)**: count function calculate total number of values present except null value.

SUM,AVG=>numeric type values

MIN,MAX,COUNT: any type of column.

**SYNTAX**:

SELECT SUM(column\_name) from table\_name where <condition>

Example:

SELECT NAME,EMAILID FROM PRODUCT\_DETAILS WHERE TFEE=(SELECT MAX(TFEE) FROM PRODUCT\_DETAILS AND DEPARTMENT=1)

ORDER BY CLAUSE: Order by clause is used with only select command.order bye clause is used at the end of the select statement .

It is used to sort selected record in acending or decending order based on a column values

SYNTAX:

select column\_name(s) from table\_name where <condition> order by column\_name asc/desc.

EXE: select \* from DAY2\_TASK1 ORDER BY TOTAL,PRODUCT\_NAME

**ALTER COMMAND**: By using alter command you an add a new column in the table,can delete a pre existing table and can modify definition or constraints of a column.

**Syntax to add new column in the table**:

alter table table\_name add column\_name data\_type(size);

**syntax to delete pre existing column**

Alter table table\_name drop column column\_name;

**Syntax to modify the column:**

Alter table table\_name alter column column\_name;

**TRUNCATE COMMAND:**  Truncate is used to reset the table.It delets all present records permanently.record deleted by truncate can not restored.

Syntax:

Truncate table table\_name;

Ex:

Truncate table tbl\_student;

Note:truncate does not remove schema .

**Drop command:** drop is used to remove database or table permanentaly.

Syntax: drop database database\_name;

Syntax to drop table:

Drop table table\_name;

**Rename command:**

It is used to rename the table name column name in tha table

Syntax for rename column name of the table:

EXEC sp\_rename 'table\_name.old\_column\_name', 'new\_column\_name', 'COLUMN';

Syntax for rename table name :

EXEC sp\_rename 'old\_table\_name', 'new\_table\_name';

**Update command:**

It is used to update the present data in the any where of the table .

Syntax:

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

**DDL (Data Definition Language)**

DDL commands are used to define and modify the database structure. Here are some common DDL commands:

1. **CREATE**: Used to create a new table or database.

**SQL**

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

BirthDate DATE

);

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

1. **ALTER**: Used to modify an existing database object, like a table.

**SQL**

1. ALTER TABLE Employees
2. ADD Email VARCHAR(100);
3. AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).
4. **DROP**: Used to delete a table or database.

**SQL**

DROP TABLE Employees;

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

**5.TRUNCATE**: Used to remove all records from a table, but not the table itself.

**SQL**

TRUNCATE TABLE Employees;

**DML (Data Manipulation Language)**

DML commands are used to manipulate the data within the database. Here are some common DML commands:

**INSERT**: Used to add new records to a table.

**SQL**

INSERT INTO Employees (EmployeeID, FirstName, LastName, BirthDate)

VALUES (1, 'John', 'Doe', '1980-01-01');

AI-generated code. Review and use carefully.

**UPDATE**: Used to modify existing records in a table.

**SQL**

UPDATE Employees

SET Email = 'john.doe@example.com'

WHERE EmployeeID = 1;

**DELETE**: Used to remove records from a table.

**SQL**

DELETE FROM Employees

WHERE EmployeeID = 1;

AI-generated code. Review and use carefully

**SELECT**: Used to retrieve data from a database.

SELECT \* FROM Employees;

ure! SQL operators are symbols and keywords used to perform operations on data. Here are the main types of SQL operators along with examples:

**1. Arithmetic Operators**

Used for mathematical operations.

* **Addition (+)**: SELECT 5 + 3;
* **Subtraction (-)**: SELECT 5 - 3;
* **Multiplication (\*)**: SELECT 5 \* 3;
* **Division (/)**: SELECT 5 / 3;
* **Modulo (%)**: SELECT 5 % 3;

**2. Comparison Operators**

Used to compare values.

* **Equal to (=)**: SELECT \* FROM Employees WHERE Age = 30;
* **Not equal to (<> or !=)**: SELECT \* FROM Employees WHERE Age <> 30;
* **Greater than (>)**: SELECT \* FROM Employees WHERE Age > 30;
* **Less than (<)**: SELECT \* FROM Employees WHERE Age < 30;
* **Greater than or equal to (>=)**: SELECT \* FROM Employees WHERE Age >= 30;
* **Less than or equal to (<=)**: SELECT \* FROM Employees WHERE Age <= 30;

**3. Logical Operators**

Used to combine multiple conditions.

* **AND**: SELECT \* FROM Employees WHERE Age > 30 AND Salary > 50000;
* **OR**: SELECT \* FROM Employees WHERE Age > 30 OR Salary > 50000;
* **NOT**: SELECT \* FROM Employees WHERE NOT Age = 30;

**4. Bitwise Operators**

Used for bitwise operations.

* **AND (&)**: SELECT 5 & 3;
* **OR (|)**: SELECT 5 | 3;

**Set Operators**

Used to combine results from multiple queries.

* **UNION**: SELECT Name FROM Employees UNION SELECT Name FROM Managers;
* **INTERSECT**: SELECT Name FROM Employees INTERSECT SELECT Name FROM Managers;
* **EXCEPT**: SELECT Name FROM Employees EXCEPT SELECT Name FROM Managers;

**6. String Operators**

Used for string operations.

* **Concatenation (+ or ||)**: SELECT 'Hello' + ' ' + 'World'; or SELECT 'Hello' || ' ' || 'World';

**7. Other Operators**

* **IN**: SELECT \* FROM Employees WHERE Age IN (25, 30, 35);
* **BETWEEN**: SELECT \* FROM Employees WHERE Age BETWEEN 25 AND 30;
* **LIKE**: SELECT \* FROM Employees WHERE Name LIKE 'J%';
* **IS NULL**: SELECT \* FROM Employees WHERE Address IS NULL;