

S.Ashifa Sherene
BE-CSE
AAMEC
2022-9615
EBEON0323764991

DBMS

Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

Importance of DBMS

A DBMS provides organizations a complete, clear view into the way data is shared, ensuring there aren't unnecessary copies of data. Not to mention, A DBMS allows organizations to enforce data security and privacy policies to reduce the risk of a data breach.

Characteristics of DBMS

1. Real World Entity
2. Self-explaining nature
3. Atomicity of Operations (Transactions)
4. Concurrent Access without Anomalies
5. Stores Any Kind of Structured Data
6. Integrity
7. Ease of Access (The DBMS Queries)
8. SQL and No-SQL Databases

9. ACID Properties

10. Security

SQL

Structured query language (SQL) is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values. You can use SQL statements to store, update, remove, search, and retrieve information from the database. You can also use SQL to maintain and optimize database performance.

Importance of SQL

Structured query language (SQL) is a popular query language that is frequently used in all types of applications. Data analysts and developers learn and use SQL because it integrates well with different programming languages. For example, they can embed SQL queries with the Java programming language to build high-performing data processing applications with major SQL database systems such as Oracle or MS SQL Server. SQL is also fairly easy to learn as it uses common English keywords in its statements.

Characteristics of SQL

1. Easy to learn
2. Wide Variety of Commands
3. Stored Procedures
4. Portable Language
5. Reusability

6. Security

DDL COMMANDS

CREATE

CREATE TABLE student (Rollno int, SName char(30), Phno int, Email varchar(30), Marks int);

ALTER

- ALTER TABLE student ADD Address varchar(50);
- ALTER TABLE student DROP COLUMN Address;
- ALTER TABLE student RENAME COLUMN Email to EmailId;
- ALTER TABLE student RENAME TO studentsInformation;

TRUNCATE

TRUNCATE TABLE student;

RENAME

- ALTER TABLE student RENAME COLUMN Email to EmailId;
- ALTER TABLE student RENAME TO studentsInformation;

DROP

DROP TABLE student;

DML COMMANDS

INSERT

- INSERT INTO student (Rollno, SName, Phno, Email, Marks) VALUES ('1','Amara', '1234512345', 'Amara@gmail.com', '98');
- INSERT INTO student (Rollno, SName, Phno, Email, Marks) VALUES ('2','Ashifa', '1234512345', 'Ashifa@gmail.com', '100');

- INSERT INTO student (Rollno, SName, Phno, Email, Marks) VALUES ('3','Zara', '1234012340', 'Zara@gmail.com', '98');

UPDATE

UPDATE student SET Marks= '90' WHERE SName= 'Zara';

DELETE

- DELETE FROM student where Phno='12341234555';
- DELETE FROM student WHERE Marks= '90';
- DELETE FROM student WHERE SName= 'Zara';

DRL COMMAND

SELECT

- SELECT * FROM student;

AGGREGATE FUNCTIONS

- SELECT COUNT (Marks) FROM student;
- SELECT SUM (Marks) FROM student WHERE name='Ashifa';
- SELECT AVG (Marks) FROM student;
- SELECT MIN (Marks) FROM student;
- SELECT MAX (Marks) FROM student;
- SELECT DISTINCT (Marks) FROM student;
- SELECT COUNT_BIG (Marks) FROM student;

TCL COMMAND

COMMIT

COMMIT TRANSACTION;

ROLLBACK

ROLLBACK;

SAVEPOINTS

- BEGIN TRANSACTION;
- SAVE TRANSACTION A;
- ROLLBACK TRANSACTION A;

JOINS

INNER JOIN

SELECT customer.customer_id, customer.name, customer.order_id,
order_table.product

FROM customer

INNER JOIN order_table

ON customer.order_id = order_table.order_id;

FULL OUTER JOIN

SELECT customer.customer_id, customer.name, customer.order_id,
order_table.product

FROM customer

FULL OUTER order_table

ON customer.order_id = order_table.order_id;

LEFT JOIN

```
SELECT customer.customer_id, customer.name, customer.order_id,  
order_table.product
```

```
FROM customer
```

```
LEFT JOIN order_table
```

```
ON customer.order_id = order_table.order_id;
```

RIGHT JOIN

```
SELECT customer.customer_id, customer.name, customer.order_id,  
order_table.product
```

```
FROM customer
```

```
RIGHT JOIN order_table
```

```
ON customer.order_id = order_table.order_id;
```

SELF JOIN

```
SELECT customer.customer_id, customer.name, customer.order_id,  
order_table.product
```

```
FROM customer, order_table
```

```
WHERE customer.order_id = order_table.order_id;
```

CROSS JOIN

```
SELECT * from customer cross join order_table;
```

