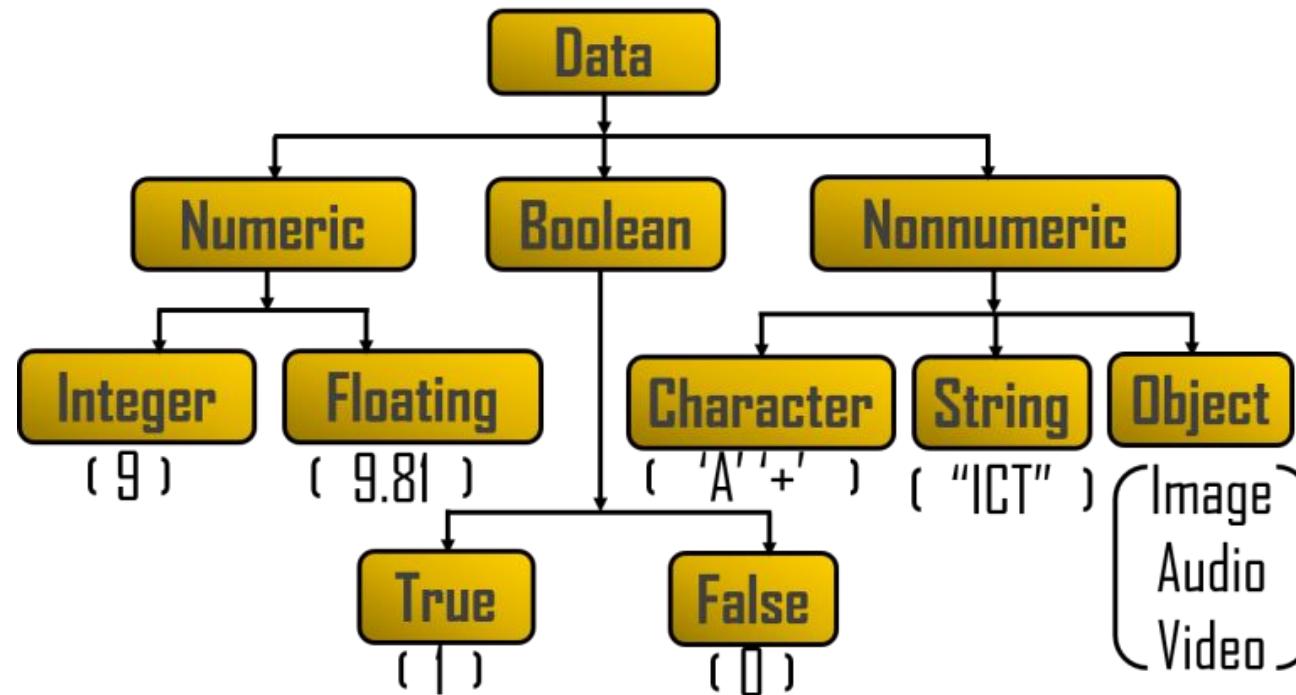


Database

Abdullah Al Maruf

Data: Data is raw, unorganized facts that need to be processed. For getting desired output in a processing data is used as input. Data can be defined as a representation of facts and it is the smallest part of information. Data is represented with the help of characters such as alphabets (A-Z, a-z), digits(0-9) or special characters (+,-,/,*,<,>,=).

Classification of data:



Information: When data is processed, organized, structured or presented in a given context so as to make it useful, it is called information. Information is the processed data on which decisions and actions are taken.

Examples of data and information: The marks of every subject of a student is called data. On the other hand, the report or result of a student by calculating the marks of all subjects is called information.



Difference between data and information:

Data	Information
→ For getting desired output in a processing data is used as input.	→ Information is the processed data on which decisions and actions are taken.
→ Data is a single concept.	→ Information is a collective concept.
→ Data doesn't give a complete idea about an object.	→ Information gives a complete idea about an object.
→ Data cannot be used directly.	→ Human being use information directly according to their needs.
→ There are some classifications of data.	→ There is no such classification of information.

Database: A database is an organized collection of related data, stored and accessed electronically. A database is used by an organization as a method of storing, managing and retrieving information. Modern databases are managed using a database management system (DBMS). It may have single or related multiple tables in a database.

Data Hierarchy: Data hierarchy is a systematic organization of data mainly in a hierarchical form. Data organization basically involves bit, byte, field, record, file, and database.

Hierarchy	Example												
Database	<p style="text-align: center;">Student Database</p> <div style="display: flex; justify-content: space-around; align-items: center;"><div style="border: 1px solid black; padding: 5px;">Basic info file</div><div style="margin: 0 20px;"></div><div style="border: 1px solid black; padding: 5px;">Tuition fees file</div><div style="margin: 0 20px;"></div><div style="border: 1px solid black; padding: 5px;">Result file</div></div>												
File	<p style="text-align: center;">Student info files</p> <table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th>Name</th><th>Section</th><th>GPA</th></tr></thead><tbody><tr><td>Monir</td><td>A</td><td>4:50</td></tr><tr><td>Kobir</td><td>B</td><td>4:60</td></tr><tr><td>Rahat</td><td>C</td><td>5:00</td></tr></tbody></table>	Name	Section	GPA	Monir	A	4:50	Kobir	B	4:60	Rahat	C	5:00
Name	Section	GPA											
Monir	A	4:50											
Kobir	B	4:60											
Rahat	C	5:00											
Record	<p style="text-align: center;">Student Record</p> <table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th>Name</th><th>Section</th><th>GPA</th></tr></thead><tbody><tr><td>Monir</td><td>A</td><td>5:00</td></tr></tbody></table>	Name	Section	GPA	Monir	A	5:00						
Name	Section	GPA											
Monir	A	5:00											
Field	<p style="text-align: center;">Student Name field</p> <div style="display: flex; align-items: center; justify-content: center;"><div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Name</div><div style="border: 1px solid black; padding: 5px;">Monir</div></div>												
Byte	01001101 (Letter M in ASCII)												
Bit	0												

Advantages of database:

- 1. Reduced data redundancy.
- 2. Reduced updating errors and increased consistency.
- 3. Greater data integrity and independence from applications programs.
- 4. Improved data access to users through use of host and query languages.
- 5. Improved data security.
- 6. Reduced data entry, storage, and retrieval costs.
- 7. Facilitated development of new applications program.

Application areas of database:

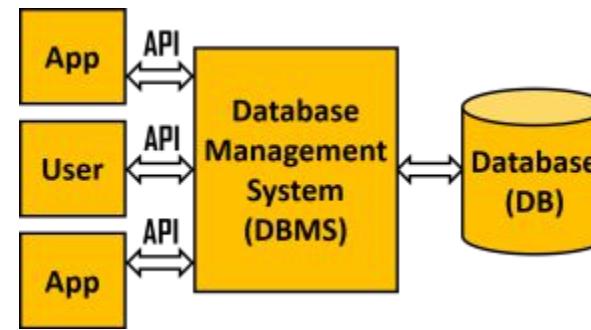
- 1. Banking
- 2. Airlines
- 3. Library
- 4. Educational Institution
- 5. Credit Card
- 6. Telecommunication
- 7. Production and distribution
- 8. Human Resource (HR)
- 9. Automated Teller Machine (ATM)
- 10. Stock/Share Market

Disadvantages of database:

- 1. Database systems are complex, difficult, and time-consuming to design.
- 2. Substantial hardware and software start-up costs.
- 3. Damage to database affects virtually all applications programs.
- 4. Extensive conversion costs in moving from a file-based system to a database system.
- 5. Initial training required for all programmers and users.

DBMS: A Database Management System (DBMS) is one kind of system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end users or application programs.

Some DBMS examples include MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, dBASE, Clipper, and FoxPro.



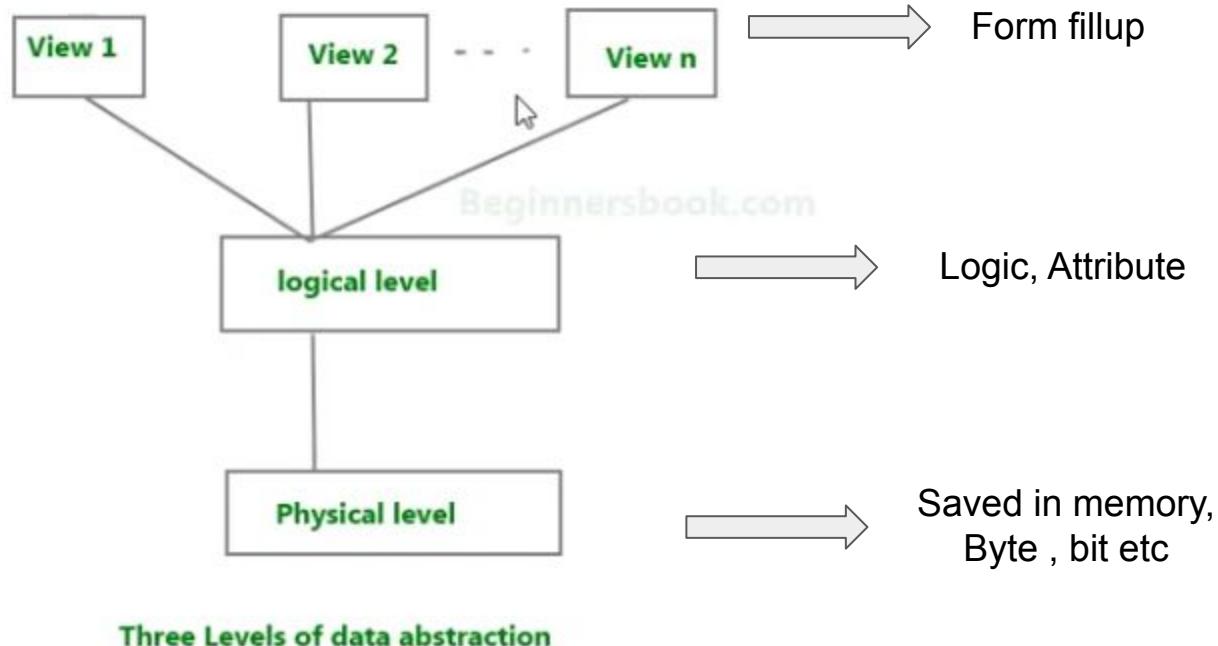
Functions of DBMS:

- 1. Data dictionary management
- 2. Data storage management
- 3. Data transformation and presentation
- 4. Security management
- 5. Multi-user access control
- 6. Backup and recovery management
- 7. Data integrity management
- 8. Database access interfaces
- 9. Database communication interfaces
- 10. Transaction management.

ডেটাবেজ ম্যানেজমেন্ট সিস্টেম এর কাজ কি কি ?

1. CREATE DATABASE
2. UPDATE DATABASE
3. PRINT DATABASE
4. BACKUP AND RECOVERY DATABASE
5. SAVE DATA
6. RETRIEVE DATA
7. DELETE DATA
8. SECURE DATA etc.

Data View



Advantages and disadvantages of DBMS:

Advantages of DBMS:

- **Better data sharing:** The DBMS helps create an environment in which end users have better access to more and better-managed data.
- **Improved data security:** A DBMS provides a framework for better enforcement of data privacy and security policies.
- **Improved data integration:** It becomes much easier to see how actions in one segment of the company affect other segments.
- **Minimized data inconsistency:** Data inconsistency exists when different versions of the same data appear in different places. The probability of data inconsistency is greatly reduced in a properly designed database.
- **Improved data access:** The DBMS makes it possible to produce quick answer to ad hoc queries.
- **Improved decision making:** The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.

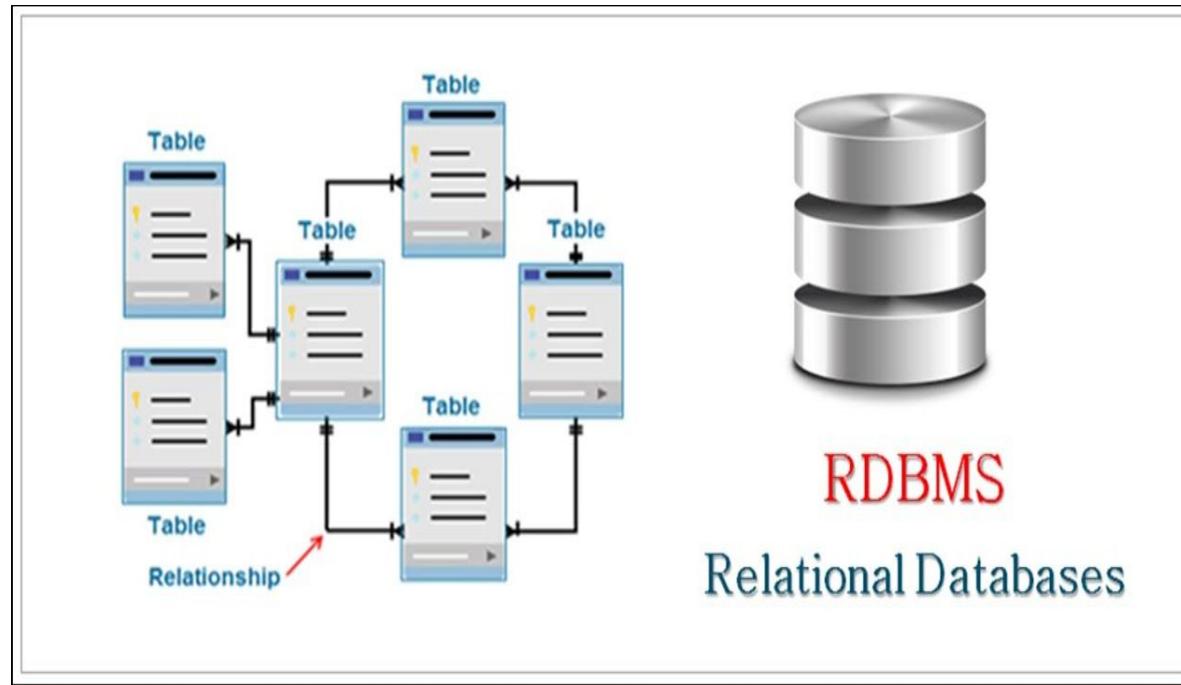
Disadvantages of DBMS:

- **Increased costs:** Database systems require sophisticated hardware and software and highly skilled personnel. The cost of maintaining the hardware, software, and personnel required to operate and manage a database system can be substantial.
- **Management Complexity:** Database systems interface with many different technologies and have a significant impact on a company's resources and culture.
- **Maintaining currency:** To maximize the efficiency of the database system, you must keep your system current. You must perform frequent updates and apply the latest patches and security measures to all components.
- **Frequent upgrade/replacement cycles:** DBMS vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new features.

RDBMS: A Relational Database management System(RDBMS) is a database management system based on the relational model introduced by E.F Codd. In relational model, data is stored in relations(tables) and is represented in form of tuples(rows).

RDBMS is used to manage Relational database. Relational database is a collection of organized set of tables related to each other, and from which data can be accessed easily. Relational Database is the most commonly used database these days.

Difference between RDBMS and DBMS: Database Management System (DBMS) is a software that is used to define, create and maintain a database and provides controlled access to the data. Relational Database Management System (RDBMS) is an advanced version of a DBMS.





A diagram illustrating a relationship between two tables. A horizontal double-headed arrow connects the 'Student Table' on the left to the 'Result Table' on the right, indicating a bidirectional association or link between the two datasets.

Roll	Name	Group
101	Ayesha	Science
102	Tamim	Science
103	Ayesha	Arts
104	Lipu	Science
105	Rabeya	Arts

Student Table

Reg No.	Roll	GPA	Section
100101	101	4.5	A
100102	102	4.5	A
100103	103	5	B
100104	104	4.7	A
100105	105	5	B

Result Table

DBMS	RDBMS
DBMS stores data as file.	RDBMS stores data in tabular form.
Data elements need to access individually.	Multiple data elements can be accessed at the same time.
No relationship between data.	Data is stored in the form of tables which are related to each other.
Normalization is not present.	Normalization is present.
DBMS does not support distributed database.	RDBMS supports distributed database.
It stores data in either a navigational or hierarchical form.	It uses a tabular structure where the headers are the column names, and the rows contain corresponding values.
It deals with small quantity of data.	It deals with large amount of data.

DBMS	RDBMS
Data redundancy is common in this model.	Keys and indexes do not allow Data redundancy.
It is used for small organization and deal with small data.	It is used to handle large amount of data.
It supports single user.	It supports multiple users.
Data fetching is slower for the large amount of data.	Data fetching is fast because of relational approach.
The data in a DBMS is subject to low security levels with regards to data manipulation.	There exists multiple levels of data security in a RDBMS.
Low software and hardware necessities.	Higher software and hardware necessities.
Examples: XML, Microsoft Access, etc.	Examples: MySQL, PostgreSQL, SQL Server, Oracle, etc.

Database Administrator (DBA): Database administrator is an individual who is responsible for the maintenance and operation of database to keep the data secure.

A database administrator's responsibilities can include the following tasks:

- 1. Installing and upgrading the database server and application tools.
- 2. Allocating system storage and planning storage requirements for the database system.
- 3. Modifying the database structure as necessary.
- 4. Enrolling users and maintaining system security.
- 5. Controlling and monitoring user access to the database.
- 6. Monitoring and optimizing the performance of the database.
- 7. Planning for backup and recovery of database information.
- 8. Backing up and restoring databases.
- 9. Generating various reports by querying from database as per need.
- 10. Managing and monitoring data replication.

- দুই বা ততোধিক টেবিলের মধ্যে যে পদ্ধতিতে সম্পর্ক স্থাপন করে টেবিলের উপর বিভিন্ন কুয়েরি চালানো হয় তাকে ডেটাবেজ রিলেশন বলে।
- টেবিলগুলোর মধ্যে কিভাবে রেকর্ডগুলো সম্পর্কযুক্ত হবে তার উপর নির্ভর করে ডেটাবেজ রিলেশনকে ৩ ভাগে ভাগ করা যায়।
 1. One-To-One
 2. One-To-Many
 3. Many-To-Many

One-To-One Database Relation

Table - A

Roll	Name	Group
101	Ayesha	Science
102	Tamim	Science
103	Ayesha	Arts

Table - B

Roll	GPA	Section
101	4.5	A
102	4.5	A
103	5	B

১। কোন একটি টেবিলের একটি রেকর্ডের সাথে যখন অন্য টেবিলের কেবল একটি রেকর্ডের সম্পর্ক স্থাপন করা হয়, তখন তাকে One-To-One Relation বলে ।

২। খুব বেশী ব্যবহার করা হয়না ।

৩। প্রাইমারী কি এর সাথে প্রাইমারী কি এর সম্পর্ক স্থাপন করা হয় ।

৪। টেবিলের ডেটার গোপনীয়তা রক্ষা করা যায় ।

Table - A

Table - B



One-To-Many (Many-To-One) Database Relation

Table - A

Roll	Name
101	Ayesha
102	Tamim
103	Ayesha

১। কোন একটি টেবিলের একটি রেকর্ডের সাথে যথন অন্য টেবিলের একাধিক রেকর্ডের সম্পর্ক স্থাপন করা হয়, তখন তাকে One-To-Many Relation বলে ।

২। সবচেয়ে বেশী ব্যবহার করা হয়।

৩। একটি টেবিলের প্রাইমারী কি এর সাথে অন্য টেবিলের ফরেন কি এর সম্পর্ক হলে এই ধরণের সম্পর্ক তৈরি হয় ।

Roll	Subject_ID	Subject_Title	Credit
101	C1001	English	3
101	C1002	Bangla	3
102	C1003	Math	3
103	C1004	ICT	3

Table - A



Table - B

Many-To-Many Database Relation

Teacher Table

Teacher_ID	Teacher_Name
T01	Rumel
T02	Sohid
T03	Alak

Course Table

Course_ID	Subject_Name
C01	English
C02	Bangla
C03	Math

Teacher_ID	Course_ID	Day
T01	C01	Mon
T01	C02	Sat
T02	C03	Tue
T03	C01	Mon

১। দুটি টেবিলের মধ্যে যদি একাধিক রেকর্ডের ম্যাচিং থাকে, তখন তাকে Many-To-Many Relation বলে।



২। এই রিলেশন তৈরির জন্য অতিরিক্ত একটি টেবিলের প্রয়োজন হয়, যাকে জংশন টেবিল বলে।

৩। দুটি টেবিলের প্রাইমারী কি জংশন টেবিলের ফরেন কি হিসাবে ব্যবহার করা হয় এই ধরণের সম্পর্কে।

Sorting

মেমো, হাইপারলিঙ্ক, OLE ফিল্ডে সর্টিং হ্যনা।

- Sorting অর্থ সাজানো।

7,15,78,32,65,11

- Sorting ২ ধরণের।

7,11,15,32,65,78

- Ascending (উর্ধ্বক্রম)

- Descending (নিম্নক্রম)

78,65,32,15,11,7

- Sorting অর্থ সাজানো।

Shain, anis, bijoy, zakir

- Sorting ২ ধরণের।

Anis, bijoy, shain, zakir

- Ascending (উর্ধ্বক্রম)

- Descending (নিম্নক্রম)

Zakir, Shain, bijoy, anis

GPA

Roll	Name	GPA
101	Shewly	4.5
102	Karim	2.5
103	Lipu	3.7
104	Ayesha	4.90

Student_table

Roll	Name	GPA
102	Karim	2.5
103	Lipu	3.7
101	Shewly	4.5
104	Ayesha	4.90

Student_table

Roll	Name	GPA
104	Ayesha	4.90
101	Shewly	4.5
103	Lipu	3.7
102	Karim	2.5

Student_table

Name

Roll	Name	GPA
101	Shewly	4.5
102	Karim	2.5
103	Lipu	3.7
104	Ayesha	4.90

Student_table

Roll	Name	GPA
102	Karim	2.5
103	Lipu	3.7
101	Shewly	4.5
104	Ayesha	4.90

Student_table

Roll	Name	GPA
104	Ayesha	4.90
101	Shewly	4.5
103	Lipu	3.7
102	Karim	2.5

Student_table

Ascending (উর্ধ্বাস্ক্রম)

Roll	Name	GPA
101	Shewly	4.5
102	Karim	2.5
103	Lipu	3.7
104	Ayesha	4.90

Student table

Roll	Name	GPA
102	Karim	2.5
103	Lipu	3.7
101	Shewly	4.5

Student table

১। সটিং হল ডেটা টেবিলের রেকর্ডগুলোকে কোন নির্ধারিত ফিল্ড অনুসারে সাজানো ।

২। সটিং এর ফলে ডেটা খুঁজে (searching) বের করা সহজ হয় ।

৩। ডেটাবেজে আলাদা স্টেড ফাইল তৈরি হয় ।

৪। সটিং এর ফলে মূল ডেটা ফাইলে রেকর্ডের ক্রমিক নং পরিবর্তন হয় ।

৫। ডেটাবেজে কোন রেকর্ড সংশোধন / সংযোজন করলে স্ট করা ফাইল আপডেট হয়না ।

৬। রেকর্ডগুলো সাজানোর জন্য তুলনামূলক বেশী সময়ের প্রয়োজন হয় ।

Indexing

Index file

Roll	Name	GPA
101	Shewly	4.5
102	Karim	2.5
103	Lipu	3.7
104	Ayesha	4.90

Student_table

Ascending (উর্ধ্বঃক্রম)

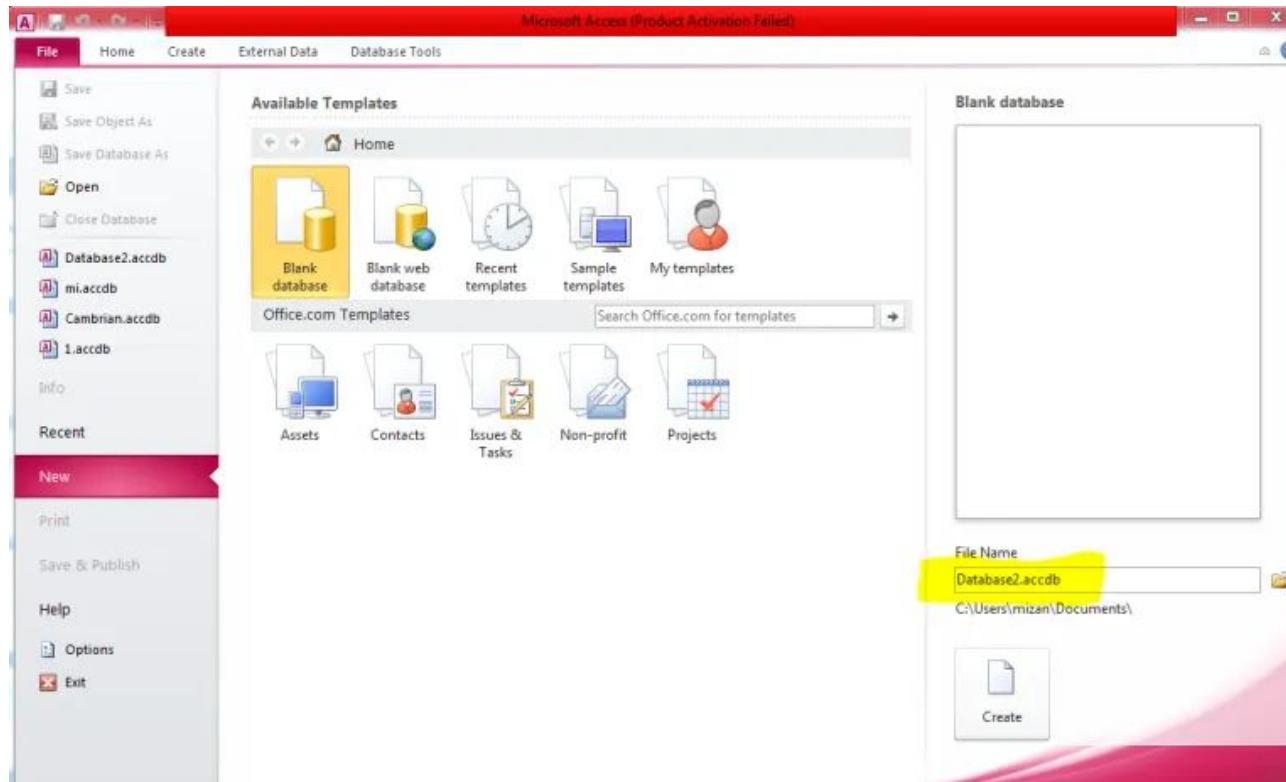
- ১। ইন্ডেক্সিং হল মূল টেবিল অপরিবর্তিত রেখে ডেটা টেবিলের রেকর্ডগুলোকে কোন নির্ধারিত ফিল্ড অনুসারে সাজানো ।
- ২। ইনডেক্স ফাইল কোন রেকর্ড থাকেনা, তবে লজিকাল অড়ার থাকে ।
- ৩। ইন্ডেক্সিং এর ফলে মূল ডেটা ফাইলে রেকর্ডের ক্রমিক নং পরিবর্তন হয় না। 
- ৪। ডেটাবেজে কোন রেকর্ড সংশোধন / সংযোজন করলে index করা ফাইল স্বয়ংক্রিয়ভাবে আপডেট হয়।
- ৫। রেকর্ডগুলো সাজানোর জন্য তুলনামূলক কম সময়ের প্রয়োজন হয় ।

Sorting vs Indexing

Sorting	Indexing
১। সটিং হল ডেটা টেবিলের রেকর্ডগুলোকে কোন নির্ধারিত ফিল্ড অনুসারে সাজানো ।	১। ইন্ডেক্সিং হল মূল টেবিল অপরিবর্তিত রেখে ডেটা টেবিলের রেকর্ডগুলোকে কোন নির্ধারিত ফিল্ড অনুসারে সাজাব্যে ।
২। সটিং এর ফলে মূল ডেটা ফাইলে রেকর্ডের ক্রমিক নং পরিবর্তন হয় ।	২। ইন্ডেক্সিং এর ফলে মূল ডেটা ফাইলে রেকর্ডের ক্রমিক নং পরিবর্তন হয় না।
৩। ডেটাবেজে কোন রেকর্ড সংশোধন / সংযোজন করলে সর্ট করা ফাইল আপডেট হয়না ।	৩। ডেটাবেজে কোন রেকর্ড সংশোধন / সংযোজন করলে index করা ফাইল আপডেট হয়।
৪। রেকর্ডগুলো সাজানোর জন্য তুলনামূলক বেশী সময়ের প্রয়োজন হয় ।	৪। রেকর্ডগুলো সাজানোর জন্য তুলনামূলক কম সময়ের প্রয়োজন হয় ।
৫। সটিং এ মেমোরি বেশী প্রয়োজন হয় ।	৫। ইন্ডেক্সিং এ মেমোরি কম প্রয়োজন হয় ।

Creating database & table and field's data type.

Creating database: For creating a database “MS Access 2010” as DBMS is used. First of all, “MS Access 2010” should be installed. After completing installation, if it is opened, the following window will be appeared.



XAMPP Control Panel v3.2.2

Modules

Service

Module

PID(s)

Port(s)

Actions



Apache

8032
5884

80, 443

Stop

Admin

Config

Logs



MySQL

5440

3306

Stop

Admin

Config

Logs



FileZilla

Start

Admin

Config

Logs



Mercury

Start

Admin

Config

Logs



Tomcat

Start

Admin

Config

Logs

 Config

 Netstat

 Shell

 Explorer

 Services

 Help

 Quit

```
09:38:36 [main]      Checking for prerequisites
09:38:36 [main]      All prerequisites found
09:38:36 [main]      Initializing Modules
09:38:36 [main]      Starting Check-Timer
09:38:36 [main]      Control Panel Ready
09:39:32 [Apache]    Attempting to start Apache app...
09:39:33 [Apache]    Status change detected: running
09:39:35 [mysql]     Attempting to start MySQL app...
09:39:36 [mysql]     Status change detected: running
```

<https://www.apachefriends.org/index.html>

You can change the database name (file name) and database location as you want from the right side. Next click the create button and database will be created.

Creating database table: A database consists of related tables. Again every table of database consists of records. A record consists of related fields. That's why it is said that field is the base of database. Before creating a table, database administrator has to fix how many fields and their data type that the table should have.

Different field's data type(Microsoft Access):

- **Short Text:** This data type is used for text or combinations of text and numbers. It can store maximum 255 characters.
- **Long Text:** This data type is used for text or combinations of text and numbers. It can store 63999 characters.
- **Number:** This data type allows any number.
- **AutoNumber:** The value of AutoNumber field gets entry automatically and usually starts at 1. It takes 4 bytes storage.
- **Date/Time:** It is used for dates and times. It takes 8 bytes storage.
- **Memo:** Memo is used for larger amounts of text. It stores up to 65,536 characters. You cannot sort a memo field. However, they are searchable.
- **Yes/No:** A logical field can be displayed as Yes/No, True/False, On/Off (equivalent to 1 and 0). Null values are not allowed in Yes/No fields. It takes 1 bit storage.
- **Currency:** This data type is used for currency. It holds up to 15 digits of whole dollars, plus 4 decimal places. You can choose which country's currency to use. It takes 8 bytes storage.
- **OLE (Object Linking and Embedding):** It can store pictures, audio, video, or other BLOBs (Binary Large Objects). It takes up to 1 GB storage.
- **Hyperlink:** It contains link or URL to other files including web pages.
- **Look up wizard:** It lets you type a list of options, which can then be chosen from a drop-down list or table. It takes 4 bytes storage.

Encryption

ডেটাকে এক জায়গা থেকে অন্য জায়গায় পাঠানোর পূর্বে মূল ফরম্যাট থেকে অন্য ফরম্যাটে
রূপান্তর করার প্রক্রিয়াকে বলে এনক্রিপশন বলে।

- ডেটা এনক্রিপ্ট করার বিভিন্ন পদ্ধতি রয়েছে।
- (যেমন - ১) সিজার কোড
- ২) DES (Data Encryption Standard)
- ৩) IDEA (International Data Encryption Algorithm)

Caesar Cipher

The **Caesar Cipher**, also known as a shift cipher, is one of the oldest and simplest forms of encrypting a message. It is a type of substitution cipher where each letter in the original message (which in cryptography is called the plaintext) is replaced with a letter corresponding to a certain number of letters shifted up or down in the alphabet.

এনক্রিপশন এর উদাহরণ

- ডেটাকে এক জায়গা থেকে অন্য জায়গায় পাঠানোর পূর্বে মূল ফরম্যাট থেকে অন্য ফরম্যাটে রূপান্তর করার প্রক্রিয়াকে বলে এনক্রিপশন বলে।
- এনক্রিপ্ট করার পূর্বের ডেটা যা পাঠ করা যায়, তাকে প্লেইন টেক্স্ট (plain text) বলে।
- এনক্রিপ্ট করার পরের ডেটা যা সহজে মানুষের পাঠযোগ্য রূপে থাকে না, তাকে সাইফার টেক্স্ট (Cipher text) বলে।
- কি (key) - গোপন কোড যা এনক্রিপ্ট বা ডিক্রিপ্ট করার জন্য প্রয়োজন।



SAMPLE ENCRYPTION AND DECRYPTION PROCESS

Encryption



.....



.....



Cipher Text

Decryption



.....



.....



Plain Text

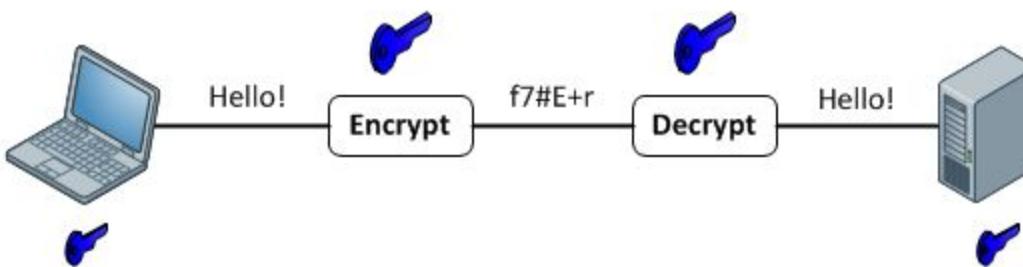
এনক্রিপশন এর প্রকারভেদ

- সাধারণত ২ ধরণের এনক্রিপশন দেখা যায় ।
 - 1) গোপন কি এনক্রিপশন / সিমেট্রিক এনক্রিপশন
 - 2) পাবলিক কি এনক্রিপশন / অ্যাসিমেট্রিক এনক্রিপশন

Symmetric Encryption

Symmetric encryption may also be referred to as *shared key* or *shared secret* encryption. In symmetric encryption, a single key is used both to encrypt and decrypt traffic.

Same
Key



Common symmetric encryption algorithms include [DES](#), [3DES](#), [AES](#), and [RC4](#). 3DES and AES are commonly used in IPsec and other types of VPNs. RC4 has seen wide deployment on wireless networks as the base encryption used by WEP and WPA version 1.

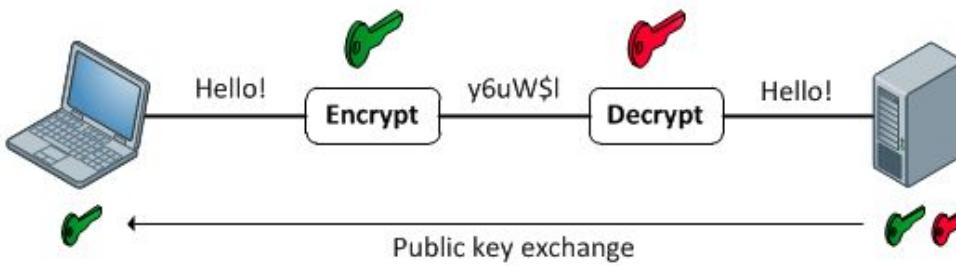
Symmetric encryption algorithms can be extremely fast, and their relatively low complexity allows for easy implementation in hardware. However, they require that all hosts participating in the encryption have already been configured with the secret key through some external means.

Asymmetric Encryption

Asymmetric encryption is also known as [public-key cryptography](#). Asymmetric encryption differs from symmetric encryption primarily in that two keys are used: one for encryption and one for decryption. The most common asymmetric encryption algorithm is [RSA](#).

Compared to symmetric encryption, asymmetric encryption imposes a high computational burden, and tends to be much slower. Thus, it isn't typically employed to protect payload data. Instead, its major strength is its ability to establish a secure channel over a nonsecure medium (for example, the Internet). This is accomplished by the exchange of public keys, which can only be used to encrypt data. The complementary private key, which is never shared, is used to decrypt.

Different Key

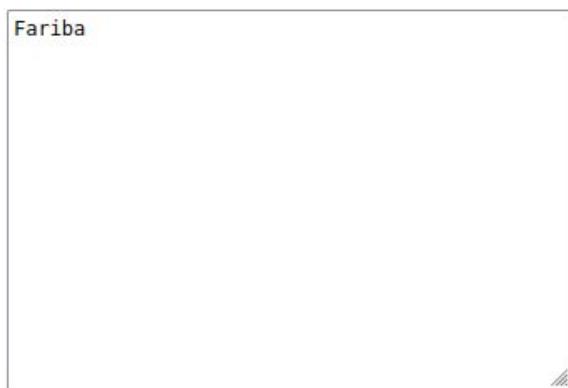


Robust encryption solutions such as IPsec implement the strengths of both symmetric and asymmetric encryption. First, two endpoints exchange public keys, which allows for the setup of a slow but secure channel. Then the two hosts decide on and exchange shared symmetric encryption keys to construct much faster symmetric encryption channels for data.

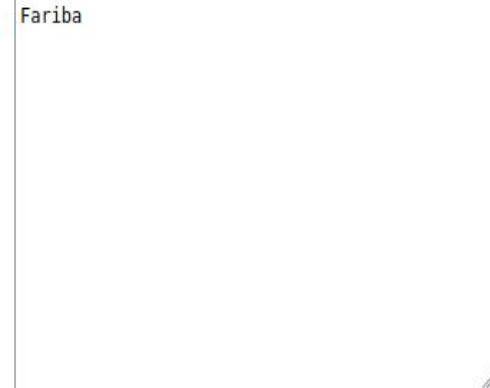
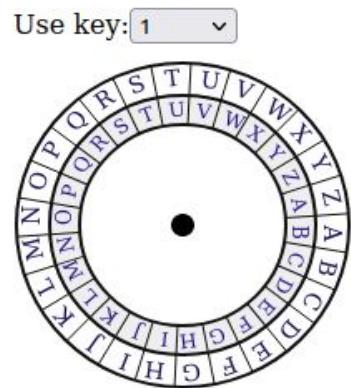
<https://learnryptography.com/Encryption/>

<http://www.unit-conversion.info/texttools/octal/>

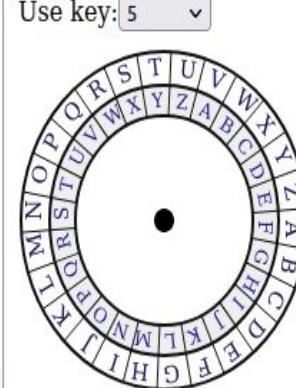
<https://www.xarg.org/tools/caesar-cipher/>



Encrypt / Decrypt



Encrypt / Decrypt



Output:

Gbsjcb

Output:

Kfwngf

VIEW

Plaintext ▾

If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out.



ENCODE DECODE

Caesar cipher ▾

SHIFT

-

7 a→h

+

ALPHABET

abcdefghijklmnopqrstuvwxyz

CASE STRATEGY

Maintain case

FOREIGN CHARS

Include Ignore

→ Encoded 163 chars

VIEW

Ciphertext ▾

Pm ol ohk hufaopun jvumpkluaphs av zhf, ol dyval pa pu ipowly, aoha pz, if zv johunpun aol vykly vm aol slaalyz vm aol hswohila, aoha uva h dvyk jvbsk il thkl vba.



Database queries, Query Languages and Different operators.

Query: A query is a request for data or information from a database table or combination of tables.

Types of query:

- 1.Select Query
- 2.Parameter Query
- 3.Crosstab Query
- 4.Action Query

Select Query: Select query is used to retrieve data from a table by selecting fields. It is the mostly used SQL query.

Parameter Query: A parameter query is one of the simplest and most useful queries that is done by taking input as a parameter from a dialog box, which is then used by the query in an expression or criteria. It is used when you want to create a query that you can use many times, but with a different value each time.

Crosstab Query: A crosstab query summarizes the data from one or more fields that are separated into groups based on one or more fields. Crosstab query is a type of select query. This query calculates a sum, average, or other aggregate function, and then groups the results.

Action Query: An action query is a query that makes changes to or moves many records in just one operation. There are four types of action queries: append, update, make-table, and delete.

- **1.Append Query**– takes the set results of a query and “appends” (or adds) them to an existing table.
- **2.Delete Query**– deletes all records in an underlying table from the set results of a query.
- **3.Make Table Query**– as the name suggests, it creates a table based on the set results of a query.
- **4.Update Query**– allows for one or more field in your table to be updated.

Query Language: Query languages are used to make queries in a database. These are some query languages-

- QUEL (Query Language)
- QBE (Query By Example)
- SQL (Structured Query Language)

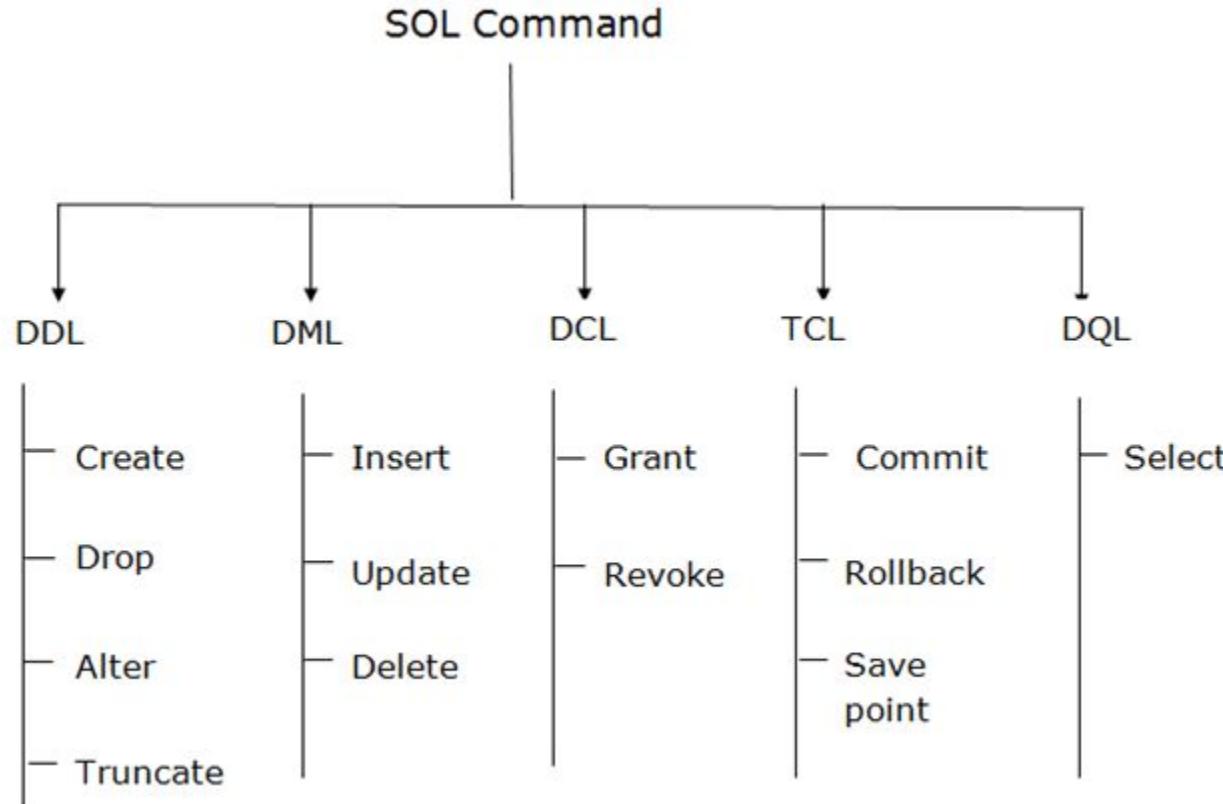
QUEL: QUEL stands for Query Language. It is a data definition and data manipulation for INGRES. INGRES stands for Interactive Graphics and Retrieval System. INGRES is a relational database management system developed by Michael Stonebraker. QUEL does not support relational algebraic operations such as intersection, minus or union. It is based on tuple calculus and does not support nested sub queries.

QBE: Query by example is a query language used in relational databases that allows users to search for information in tables and fields by providing a simple user interface where the user will be able to input an example of the data that he or she wants to access. QBE was created by Moshe Zloof at IBM in the 1970s in parallel to SQL's development. It is a graphical query language where users can input commands into a table like conditions and example elements. The principle of QBE is that it is merely an abstraction between the user and the real query that the database system will receive. In the background, the user's query is transformed into a database manipulation language form such as SQL, and it is this SQL statement that will be executed in the background.

SQL: SQL stands for Structured Query Language. SQL is a standard language for accessing and manipulating databases. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS(MySql, Oracle, Infomix, Sybase, MS Access) use SQL as the standard database query language. SQL is used to perform all types of data operations in RDBMS. Although SQL is often described as, and to a great extent is, a declarative language (4GL). It also includes procedural elements. SQL consists of many types of statements, which may be informally classed as sub-languages, commonly:

- 1.Data Definition Language (DDL)
- 2.Data Manipulation Language (DML)
- 3.Transaction Control Language(TCL)
- 4.Data Control Language (DCL)
- 5.Data query language (DQL)

Types of SQL Command:



Data Definition Language (DDL): This includes changes to the structure of the table like creation of table, altering table, deleting a table etc. All DDL commands are auto-committed. That means it saves all the changes permanently in the database.

Here are some commands that come under DDL:

CREATE- It is used to create a new table in the database. See the following syntax-

```
CREATE TABLE table_name (column_name data_types[,...]);
```

ALTER- It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute. See the following syntax-

```
ALTER TABLE table_name ADD column_name COLUMN-definition;
```

DROP- It is used to delete both the structure and record stored in the table. See the following syntax-

```
DROP TABLE table_name;
```

TRUNCATE- It is used to delete all the rows from the table and free the space containing the table. See the following syntax-

```
TRUNCATE TABLE table_name;
```

RENAME- It is used to rename a table.

Data Manipulation Language (DML): DML commands are used for manipulating the data stored in the table and not the table itself. DML commands are not auto-committed. It means changes are not permanent to database, they can be rolled back.

Here are some commands that come under DML:

INSERT – It is used to insert data into the row of a table. See the following syntax-

```
INSERT INTO table_name(col1, col2, col3,... col N) VALUES (value1, value2, value3, .... valueN);
```

UPDATE- This command is used to update or modify the value of a column in the table. See the following syntax-

```
UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [WHERE CONDITION]
```

DELETE- It is used to remove one or more row from a table. See the following syntax-

```
DELETE FROM table_name [WHERE condition];
```

Transaction Control Language(TCL): These commands are to keep a check on other commands and their affect on the database. TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only. These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

COMMIT- It is used to save all the transactions to the database.

```
COMMIT;
```

ROLLBACK- Rollback command is used to undo transactions that have not already been saved to the database.

```
ROLLBACK;
```

SAVEPOINT- It is used to roll the transaction back to a certain point without rolling back the entire transaction.

```
SAVEPOINT SAVEPOINT_NAME;
```

Data Control Language (DCL): Data control language are the commands to grant and take back authority from any database user.

Here are some commands that come under DCL:

GRANT- It is used to give user access privileges to a database.

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

REVOKE- It is used to take back permissions from the user.

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

Data query language (DQL): Data query language is used to fetch data from tables based on conditions that we can easily apply. It uses only one command:

SELECT- It is used to select the attribute based on the condition described by WHERE clause.

```
SELECT column_name/* FROM table_name WHERE conditions;
```

Database Operators: An operator is a reserved word or a character used primarily in an SQL statement's WHERE clause to perform operation(s), such as comparisons and arithmetic operations. Different types of operators-

Arithmetic Operators- You use the arithmetic operators to calculate a value from two or more numbers or to change the sign of a number from positive to negative or vice versa.

Data Definition Language (DDL):

Syntax of SQL Command for Creating Database:

```
CREATE DATABASE database_name;
```

SQL Command for creating a database named db_sagc

```
CREATE DATABASE db_sagc;
```

Syntax of SQL Command for Creating Table:

```
CREATE TABLE table_name  
( column_name_1 data_type(size),  
  column_name_2 data_type(size),  
  .....  
  column_name_n data_type(size)  
);
```

Operator	Purpose	Example
+	Sum two numbers.	[Subtotal]+[SalesTax]
-	Find the difference between two numbers	[Price]-[Discount]
*	Multiply two numbers.	[Quantity]*[Price]
/	Divide the first number by the second number.	[Total]/[ItemCount]
\	Round both numbers to integers, divide the first number by the second number, and then truncate the result to an integer.	[Registered]\[Rooms]
Mod	Divide the first number by the second number, and then return only the remainder.	[Registered] Mod [Rooms]
[^]	Raise a number to the power of an exponent.	Number ^ Exponent

Comparison Operators- You use the comparison operators to compare values and return a result that is True, False, or Null.

Operator	Purpose	Example
<	Returns True if the first value is less than the second value.	Value1 < Value2
<=	Returns True if the first value is less than or equal to the second value.	Value1 <= Value2
>	Returns True if the first value is greater than the second value.	Value1 > Value2
>=	Returns True if the first value is greater than or equal to the second value.	Value1 >= Value2
=	Returns True if the first value is equal to the second value.	Value1 = Value2
<>	Returns True if the first value is not equal to the second value.	Value1 <> Value2

Logical Operators- You use the logical operators to combine two Boolean values and return a true, false, or null result. Logical operators are also referred to as Boolean operators.

Operator	Purpose	Example
And	Returns True when Expr1 and Expr2 are true.	Expr1 And Expr2
Or	Returns True when either Expr1 or Expr2 is true.	Expr1 Or Expr2
Not	Returns True when Expr is not true.	Not Expr
Eqv	Returns True when both Expr1 and Expr2 are true, or when both Expr1 and Expr2 are false.	Expr1 Eqv Expr2
Xor	Returns True when either Expr1 is true or Expr2 is true, but not both.	Expr1 Xor Expr2

Concatenation Operators- You use the concatenation operators to combine two text values into one.

Operator	Purpose	Example
&	Combines two strings to form one string.	string1 & string2
+	Combines two strings to form one string and propagates null values (if one value is Null, the entire expression evaluates to Null).	string1 + string2

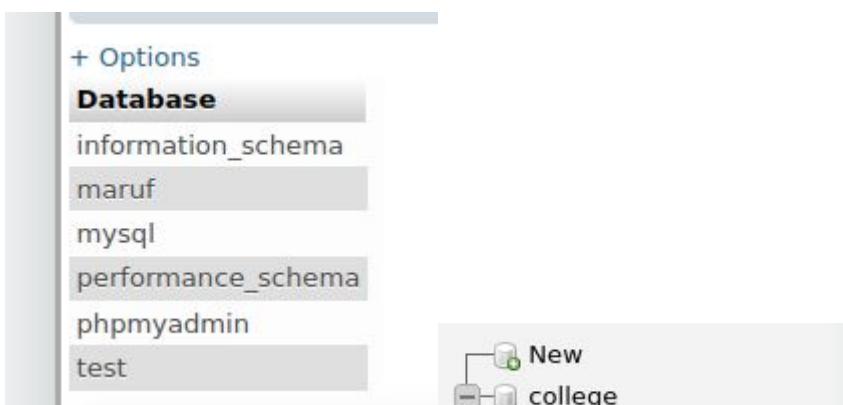
Special operators- You use the special operators to return a True or False result as described in the following table.

Operator	Purpose	Example
IsNull or IsNotNull	Determines whether a value is Null or Not Null.	Field1 Is Not Null
Like "pattern"	Matches string values by using the wildcard operators ? and *.	Field1 Like "instruct*"
Between val1 And val2	Determines whether a numeric or date value is found within a range.	Field1 Between 1 And 10 - OR - Field1 Between #07-01-07# And #12-31-07#
In(val1,val2...)	Determines whether a value is found within a set of values.	Field1 In ("red","green","blue") - OR - Field1 In (1,5,7,9)

List of databases

Syntax →

```
SHOW DATABASES;
```



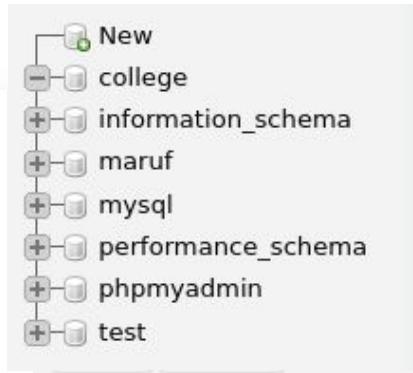
Create a database

Syntax →

```
CREATE DATABASE database_name;
```

Example →

```
CREATE DATABASE college;
```



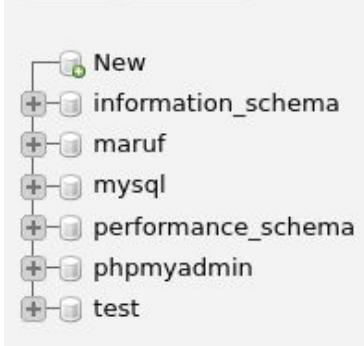
Delete a database

Syntax →

```
DROP DATABASE database_name;
```

Example →

```
DROP DATABASE college;
```



- Numeric data type

• SMALLINT

• INT

• BIGINT

• FLOAT

• DOUBLE(M,D)

• DECIMAL(M,D)

- Character Type

• Char(m)

• Varchar(m)

• text

Roll	Name	Gender	Age	GPA	City
101	Maruf	Male	21	4.52	Dhaka
102	Arif	Male	17	4.56	Cumilla
103	Mahbuba	Female	18	4.68	Sylhet
104	Shimul	Male	16	4.57	Barishal
105	Sharmin	Female	18	4.63	Rangpur
106	Nayan	Male	19	4.89	Nowakhali
107	Murad	Male	20	4.78	Chittagong
108	Anika	Female	21	4.43	Sylhet

- Data and Time data Type

• YEAR (YYYY)

• DATE (YYYY-MM-DD)

• TIME(HH-MM-SS)

S
Y
N
T
A
X

- টেবিল তৈরির জন্য CREATE TABLE statement ব্যবহার করা হয়।
 - Basic Syntax
- CREATE TABLE** table_name
(
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
...
columnN data_type(size)
);

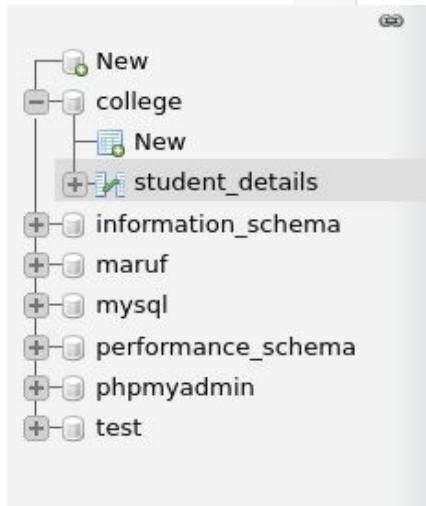
```
1 CREATE TABLE student
2 (
3     Roll int(5),
4     Name varchar(20),
5     Gender varchar(10),
6     Age int(5),
7     GPA double(3,2),
8     City varchar(15),
9
10    PRIMARY KEY(Roll)
11 );
```

rename a table →

RENAME TABLE student TO student_details;

delete a table →

DROP TABLE student_details;



insert data →

INSERT INTO table_name(column1, column2, column3,...,columnN)

VALUES (value1, value2, value3, ...valueN);

Example:

INSERT INTO student_details(Roll, Name, Gender, Age, GPA, City)

VALUES(101, 'Rahim', 'Male', 18, 4.89, 'Dhaka');

	Roll	Name	Gender	Age	GPA	City
1	101	Maruf	Male	21	4.52	Dhaka

adding multiple row at the same time

```
1 INSERT INTO student_details  
2 VALUES  
3     (102, 'Arif', 'Male', 17, 4.56, 'Cumilla'),  
4     (103, 'Mahabuba', 'Female', 18, 4.68, 'Sylhet'),  
5     (104, 'Shimul', 'Male', 16, 4.57, 'Barishal'),  
6     (105, 'Sharmin', 'Female', 18, 4.63, 'Rangpur'),  
7     (106, 'Nayan', 'Male', 19, 4.89, 'Nowakhali'),  
8     (107, 'Murad', 'Male', 20, 4.78, 'Chittagong'),  
9     (108, 'Anika', 'Female', 21, 4.43, 'Sylhet')  
10 ;
```

Roll	Name	Gender	Age	GPA	City
101	Maruf	Male	21	4.52	Dhaka
102	Arif	Male	17	4.56	Cumilla
103	Mahabuba	Female	18	4.68	Sylhet
104	Shimul	Male	16	4.57	Barishal
105	Sharmin	Female	18	4.63	Rangpur
106	Nayan	Male	19	4.89	Nowakhali
107	Murad	Male	20	4.78	Chittagong
108	Anika	Female	21	4.43	Sylhet

- DDL -> CREATE, ALTER, DROP
 - DML -> INSERT, SELECT, UPDATE, DELETE
-
- SELECT statement এর সাহায্যে টেবিল থেকে প্রযোজনীয় তথ্য খুঁজে পাওয়া যায়।

Syntax

```
SELECT column_list  
FROM table_name;
```

Now we want to see Name column from the table

```
1 SELECT name FROM student_details;
```

+ options

	name
<input type="checkbox"/>	Edit Copy Delete Maruf
<input type="checkbox"/>	Edit Copy Delete Arif
<input type="checkbox"/>	Edit Copy Delete Mahabuba
<input type="checkbox"/>	Edit Copy Delete Shimul
<input type="checkbox"/>	Edit Copy Delete Sharmin
<input type="checkbox"/>	Edit Copy Delete Nayan
<input type="checkbox"/>	Edit Copy Delete Murad
<input type="checkbox"/>	Edit Copy Delete Anika

Now we want to see gpa, name, age column from the table

```
1 SELECT Name, GPA, Age FROM student_details;
```

	Name	GPA	Age
<input type="checkbox"/>	Edit Copy Delete Maruf	4.52	21
<input type="checkbox"/>	Edit Copy Delete Arif	4.56	17
<input type="checkbox"/>	Edit Copy Delete Mahabuba	4.68	18
<input type="checkbox"/>	Edit Copy Delete Shimul	4.57	16
<input type="checkbox"/>	Edit Copy Delete Sharmin	4.63	18
<input type="checkbox"/>	Edit Copy Delete Nayan	4.89	19
<input type="checkbox"/>	Edit Copy Delete Murad	4.78	20
<input type="checkbox"/>	Edit Copy Delete Anika	4.43	21

Now we want to see the full table

Syntax →

```
SELECT * FROM student_details;
```

	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	Edit  Copy 	101 Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	Edit  Copy 	102 Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>	Edit  Copy 	103 Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	Edit  Copy 	104 Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>	Edit  Copy 	105 Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>	Edit  Copy 	106 Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>	Edit  Copy 	107 Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>	Edit  Copy 	108 Anika	Female	21	4.43	Sylhet

DISTINCT (duplicate value thakle segula bad diye dibe)

Syntax →

```
SELECT DISTINCT City  
FROM student_details;
```

Roll	Name	Gender	Age	GPA	City
101	Maruf	Male	21	4.52	Dhaka
102	Arif	Male	17	4.56	Cumilla
103	Mahbuba	Female	18	4.68	Sylhet
104	Shimul	Male	16	4.57	Barishal
105	Sharmin	Female	18	4.63	Rangpur
106	Nayan	Male	19	4.89	Nowakhali
107	Murad	Male	20	4.78	Chittagong
108	Anika	Female	21	4.43	Sylhet

Example

```
1 SELECT DISTINCT City, Age FROM student_details;
```

	City	Age
<input type="checkbox"/>	Dhaka	21
<input type="checkbox"/>	Cumilla	17
<input type="checkbox"/>	Sylhet	18
<input type="checkbox"/>	Barishal	16
<input type="checkbox"/>	Rangpur	18
<input type="checkbox"/>	Nowakhali	19
<input type="checkbox"/>	Chittagong	20
<input type="checkbox"/>	Sylhet	21

```
1 SELECT DISTINCT Age FROM student_details;
```

Age
21
17
18
16
19
20

LIMIT Koyta row dekhabe

```
1 SELECT * FROM student_details  
2 LIMIT 5;
```

	Edit	Copy	Delete	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>				101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>				102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>				103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>				104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>				105	Sharmin	Female	18	4.63	Rangpur

```
1 SELECT * FROM student_details  
2 LIMIT 2,5;
```

	Edit	Copy	Delete	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>				103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>				104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>				105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>				106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>				107	Murad	Male	20	4.78	Chittagong

ORDER BY

Ascending Order

```
1 SELECT * FROM student_details  
2 ORDER BY Name;
```

	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>  Edit  Copy  Delete	108	Anika	Female	21	4.43	Sylhet
<input type="checkbox"/>  Edit  Copy  Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>  Edit  Copy  Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>  Edit  Copy  Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>  Edit  Copy  Delete	107	Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>  Edit  Copy  Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>  Edit  Copy  Delete	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>  Edit  Copy  Delete	104	Shimul	Male	16	4.57	Barishal

Descending Order

```
1 SELECT * FROM student_details  
2 ORDER BY Name DESC;
```

	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>  Edit  Copy  Delete	104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>  Edit  Copy  Delete	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>  Edit  Copy  Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>  Edit  Copy  Delete	107	Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>  Edit  Copy  Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>  Edit  Copy  Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>  Edit  Copy  Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>  Edit  Copy  Delete	108	Anika	Female	21	4.43	Sylhet

Order by GPA

```
1 SELECT * FROM student_details  
2 ORDER BY GPA DESC;
```

	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>	107	Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>	104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	108	Anika	Female	21	4.43	Sylhet

arithmetic operator

```
select 5 + 2;  
ans: 7
```

```
select 5 - 2;  
ans: 3
```

```
select 5 * 2;  
ans: 10
```

```
select 5 / 2;  
ans: 2.5000
```

```
select 5 % 2;  
ans: 1
```

WHERE

```
1 SELECT City FROM student_details  
2 WHERE Gender = 'Female';
```

City				
	Edit	Copy	Delete	
Sylhet	<input type="checkbox"/>			
Rangpur	<input type="checkbox"/>			
Sylhet	<input type="checkbox"/>			

```
1 SELECT * FROM student_details  
2 WHERE Age = 18;
```

	Roll	Name	Gender	Age	GPA	City
	Edit	Copy	Delete			
103	Mahabuba	Female	18	4.68	Sylhet	
105	Sharmin	Female	18	4.63	Rangpur	

Between, AND

```
1 SELECT * FROM student_details  
2 WHERE Roll BETWEEN 103 AND 107;
```

	Roll	Name	Gender	Age	GPA	City
	Edit	Copy	Delete			
103	Mahabuba	Female	18	4.68	Sylhet	
104	Shimul	Male	16	4.57	Barishal	
105	Sharmin	Female	18	4.63	Rangpur	
106	Nayan	Male	19	4.89	Nowakhali	
107	Murad	Male	20	4.78	Chittagong	

Another Example

```
1 SELECT * FROM student_details  
2 WHERE GPA BETWEEN 4.70 AND 5.00;
```

+ Options

	← T →	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	Edit Copy Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>	Edit Copy Delete	107	Murad	Male	20	4.78	Chittagong

Not equal to Operator

```
1 SELECT * FROM student_details  
2 WHERE Roll != 105;
```

	← T →	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	Edit Copy Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	Edit Copy Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>	Edit Copy Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	Edit Copy Delete	104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>	Edit Copy Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>	Edit Copy Delete	108	Anika	Female	21	4.43	Sylhet

Using 2 not equal to operator at the same time

```
1 SELECT * FROM student_details  
2 WHERE Roll != 105 AND Roll != 107;
```

+ Options												
	← T →	▼	Roll	Name	Gender	Age	GPA	City				
<input type="checkbox"/>		Edit		Copy		Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>		Edit		Copy		Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>		Edit		Copy		Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>		Edit		Copy		Delete	104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>		Edit		Copy		Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>		Edit		Copy		Delete	108	Anika	Female	21	4.43	Sylhet

OR Operator

```
1 SELECT * FROM student_details  
2 WHERE Gender = "Male" or Age >= 18;
```

+ Options

	← T →	▼	Roll	Name	Gender	Age	GPA	City	
<input type="checkbox"/>	 Edit	 Copy	 Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	 Edit	 Copy	 Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>	 Edit	 Copy	 Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	 Edit	 Copy	 Delete	104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>	 Edit	 Copy	 Delete	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>	 Edit	 Copy	 Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>	 Edit	 Copy	 Delete	107	Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>	 Edit	 Copy	 Delete	108	Anika	Female	21	4.43	Sylhet

```
1 SELECT * FROM student_details  
2 WHERE Gender = "Female" OR GPA >= 4.50;
```

+ Options

	← ↑ →	▼	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	 Edit	 Copy	 Delete	101 Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	 Edit	 Copy	 Delete	102 Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>	 Edit	 Copy	 Delete	103 Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	 Edit	 Copy	 Delete	104 Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>	 Edit	 Copy	 Delete	105 Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>	 Edit	 Copy	 Delete	106 Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>	 Edit	 Copy	 Delete	107 Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>	 Edit	 Copy	 Delete	108 Anika	Female	21	4.43	Sylhet

And Operator

```
1 SELECT * FROM student_details  
2 WHERE Gender = "Male" AND Age >= 18;
```

	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/> Edit Copy Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/> Edit Copy Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/> Edit Copy Delete	107	Murad	Male	20	4.78	Chittagong

```
1 SELECT * FROM student_details  
2 WHERE Gender = "Female" AND GPA >= 4.50;
```

+ Options

	← T →	▼	Roll	Name	Gender	Age	GPA	City	
<input type="checkbox"/>	 Edit	 Copy	 Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	 Edit	 Copy	 Delete	105	Sharmin	Female	18	4.63	Rangpur

Using both AND | OR

```
1 SELECT * FROM student_details  
2 WHERE City = 'Sylhet' AND (Gender = 'Female' OR GPA = 4.50);
```

	← T →	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	Edit Copy Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	Edit Copy Delete	108	Anika	Female	21	4.43	Sylhet

```
1 SELECT * FROM student_details  
2 WHERE Age > 18 AND (Gender = 'Male' OR GPA <= 4.00);
```

+	OPTIONS	← T →	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	Edit Copy Delete	101	Maruf	Male	21	4.52	Dhaka	
<input type="checkbox"/>	Edit Copy Delete	106	Nayan	Male	19	4.89	Nowakhali	
<input type="checkbox"/>	Edit Copy Delete	107	Murad	Male	20	4.78	Chittagong	

IN OPERATOR

```
1 SELECT * FROM student_details  
2 WHERE City = 'Dhaka' OR City = 'Cumilla' OR City = 'Sylhet';
```

+ Options

	← T →	▼	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	 Edit	 Copy	 Delete	101 Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	 Edit	 Copy	 Delete	102 Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>	 Edit	 Copy	 Delete	103 Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	 Edit	 Copy	 Delete	108 Anika	Female	21	4.43	Sylhet

```
1 SELECT * FROM student_details  
2 WHERE City IN('Dhaka','Cumilla','Sylhet');
```

		Roll	Name	Gender	Age	GPA	City		
<input type="checkbox"/>	Edit	Copy	Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	Edit	Copy	Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>	Edit	Copy	Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	Edit	Copy	Delete	108	Anika	Female	21	4.43	Sylhet

NOT IN OPERATOR

```
1 SELECT * FROM student_details  
2 WHERE City NOT IN('Dhaka','Cumilla','Sylhet');
```

+ Options

	← ↑ →	▼	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	 Edit	 Copy	 Delete	104	Shimul	Male	16	4.57 Barishal
<input type="checkbox"/>	 Edit	 Copy	 Delete	105	Sharmin	Female	18	4.63 Rangpur
<input type="checkbox"/>	 Edit	 Copy	 Delete	106	Nayan	Male	19	4.89 Nowakhali
<input type="checkbox"/>	 Edit	 Copy	 Delete	107	Murad	Male	20	4.78 Chittagong

LIKE OPERATOR

Logical Operator

AND
OR
IN
NOT
LIKE

```
1 SELECT * FROM student_details  
2 WHERE Name LIKE 'M%';
```

+ Options								
	← T →	▼	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	 Edit	 Copy	 Delete	101	Maruf	Male	21	4.52
<input type="checkbox"/>	 Edit	 Copy	 Delete	103	Mahabuba	Female	18	4.68
<input type="checkbox"/>	 Edit	 Copy	 Delete	107	Murad	Male	20	4.78

```
1 SELECT * FROM student_details  
2 WHERE Name LIKE 'M%' OR Name LIKE 'S%';
```

+ Options

	← T →	▼	Roll	Name	Gender	Age	GPA	City	
<input type="checkbox"/>	 Edit	 Copy	 Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	 Edit	 Copy	 Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	 Edit	 Copy	 Delete	104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>	 Edit	 Copy	 Delete	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>	 Edit	 Copy	 Delete	107	Murad	Male	20	4.78	Chittagong

+ Options

	← T →	▼	Roll	Name	Gender	Age	GPA	City				
<input type="checkbox"/>		Edit		Copy		Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>		Edit		Copy		Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>		Edit		Copy		Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>		Edit		Copy		Delete	104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>		Edit		Copy		Delete	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>		Edit		Copy		Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>		Edit		Copy		Delete	107	Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>		Edit		Copy		Delete	108	Anika	Female	21	4.43	Sylhet

```
1 SELECT * FROM student_details  
2 WHERE Name LIKE '%F';
```

+ Options

	← T →	▼	Roll	Name	Gender	Age	GPA	City				
<input type="checkbox"/>		Edit		Copy		Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>		Edit		Copy		Delete	102	Arif	Male	17	4.56	Cumilla

```
1 SELECT * FROM student_details  
2 WHERE Name LIKE '%a%';
```

+ Options

	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>  Edit  Copy  Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>  Edit  Copy  Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>  Edit  Copy  Delete	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>  Edit  Copy  Delete	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>  Edit  Copy  Delete	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>  Edit  Copy  Delete	107	Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>  Edit  Copy  Delete	108	Anika	Female	21	4.43	Sylhet

```
1 SELECT * FROM student_details  
2 WHERE Name LIKE '%r%';
```

	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>  Edit  Copy  Delete	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>  Edit  Copy  Delete	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>  Edit  Copy  Delete	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>  Edit  Copy  Delete	107	Murad	Male	20	4.78	Chittagong

1st letter ki ache seta matter na but 2nd letter h hote hobe and pore ja iccha thakuk arokom caile...

```
1 SELECT * FROM student_details  
2 WHERE Name LIKE '_h%';
```

+ Options

	← →	▼	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	 Edit	 Copy	 Delete	104	Shimul	Male	16	4.57 Barishal
<input type="checkbox"/>	 Edit	 Copy	 Delete	105	Sharmin	Female	18	4.63 Rangpur

Bangla type pari na :)

Last er ager word a thaka lagbe amon caile

```
1 SELECT * FROM student_details  
2 WHERE Name LIKE '%a_';
```

+ Options

	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	107	Murad	Male	20	4.78	Chittagong

+ Options

	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	Roll	Name	Gender	Age	GPA	City
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	101	Maruf	Male	21	4.52	Dhaka
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	102	Arif	Male	17	4.56	Cumilla
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	103	Mahabuba	Female	18	4.68	Sylhet
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	104	Shimul	Male	16	4.57	Barishal
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	105	Sharmin	Female	18	4.63	Rangpur
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	106	Nayan	Male	19	4.89	Nowakhali
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	107	Murad	Male	20	4.78	Chittagong
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	108	Anika	Female	21	4.43	Sylhet

Output dekhar somoy Column er nam onno kisu dekhte caile...

```
1 SELECT Roll AS id, Name, Age, GPA  
2 FROM student_details;
```

+ Options

				id	Name	Age	GPA
<input type="checkbox"/>				101	Maruf	21	4.52
<input type="checkbox"/>				102	Arif	17	4.56
<input type="checkbox"/>				103	Mahabuba	18	4.68
<input type="checkbox"/>				104	Shimul	16	4.57
<input type="checkbox"/>				105	Sharmin	18	4.63
<input type="checkbox"/>				106	Nayan	19	4.89
<input type="checkbox"/>				107	Murad	20	4.78
<input type="checkbox"/>				108	Anika	21	4.43

Want to see the description of the table?
And also what is the primary key of the selected table?

```
1 DESCRIBE student_details;
```

Want to change the type of any column?

```
ALTER TABLE student_details MODIFY Name  
CHAR(30) NOT NULL;
```

Field	Type	Null	Key	Default	Extra
Roll	int(5)	NO	PRI	NULL	
Name	varchar(20)	YES		NULL	
Gender	varchar(10)	YES		NULL	
Age	int(5)	YES		NULL	
GPA	double(3,2)	YES		NULL	
City	varchar(15)	YES		NULL	

Field	Type	Null	Key	Default	Extra
Roll	int(5)	NO	PRI	NULL	
Name	char(30)	NO		NULL	
Gender	varchar(10)	YES		NULL	
Age	int(5)	YES		NULL	
GPA	double(3,2)	YES		NULL	
City	varchar(15)	YES		NULL	

SQL CONSTRAINTS

Constraints

- **PRIMARY KEY** – Ensures that a column has a unique value for each record.
- **NOT NULL** – Enforces that every record has a value for the column
- **DEFAULT** – Specifies a value for the column when a value is not included
- **UNIQUE** – Enforces that each column uses distinct and unique values
- **CHECK** – Enforces specific rules that a column must follow
- **FOREIGN KEY** – Enforces the child relationship with a parent table

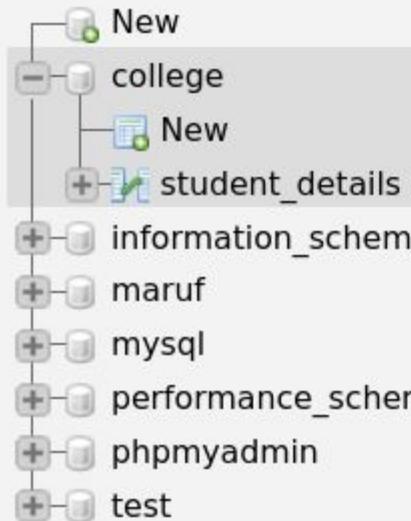
```
CREATE TABLE HR.Employees
(EmpID int IDENTITY PRIMARY KEY,
 FirstName varchar(15) NOT NULL,
 LastName varchar(20) NOT NULL,
 JobTitle varchar(20) NOT NULL,
 HireDate date DEFAULT GETDATE(),
 BirthDate date NULL,
 PhoneNumber varchar(15) UNIQUE,
 DeptCode tinyint)
```

Run SQL query/queries on table college.Teacher

```
1 INSERT INTO Teacher (id, Name)
2 VALUES
3     (1000, 'Moshiur Rahman'),
4     (1001, 'Sumit Saha')
5 ;
```

The screenshot shows the MySQL Workbench interface. On the left, a tree view displays various databases: 'New', 'college' (which contains 'student_details' and 'Teacher' tables), 'New', 'information_schema', 'maruf', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. The 'Teacher' table under the 'college' database is currently selected, indicated by a grey background. On the right, a large text area titled 'Run SQL query/queries on table college.Teacher' contains the provided SQL code. The code inserts two rows into the 'Teacher' table with IDs 1000 and 1001, names 'Moshiur Rahman' and 'Sumit Saha' respectively, and ends with a semicolon.

Recent | Favorites



Run SQL query/queries on database college: [?](#)

```
1 CREATE TABLE Teacher
2 (
3     ID int NOT NULL AUTO_INCREMENT,
4     Name varchar(15) NOT NULL,
5     Salary double(6,3) NOT NULL,
6     PRIMARY KEY(ID)
7 );
```

```
1 INSERT INTO Teacher(ID, Name, Salary)
2 VALUES
3 (1000, 'Moshiur Rahman', 35000.00);
```

	← T →	ID	Name	Salary
<input type="checkbox"/>	Edit Copy Delete	1000	Moshiur Rahman	35000.00

```
1 INSERT INTO Teacher(Name, Salary)
2 VALUES
3 ('Anisur Rahman', 45000.00);
```

+ Options	← T →	ID	Name	Salary
<input type="checkbox"/>	Edit Copy Delete	1000	Moshiur Rahman	35000.00
<input type="checkbox"/>	Edit Copy Delete	1001	Anisur Rahman	45000.00

```
1 DESCRIBE Teacher;
```

+ Options

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Name	varchar(15)	NO		NULL	
Salary	double(10,2)	NO		NULL	

```
1 ALTER TABLE Teacher MODIFY Salary Double(10,2) NULL;
```

```
1 DESCRIBE Teacher;
```

+ Options

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Name	varchar(15)	NO		NULL	
Salary	double(10,2)	YES		NULL	

```
1 INSERT INTO Teacher(Name)
2 VALUES
3 ('Moshiur Rahman'),
4 ('Dev Ed');
```

	ID	Name	Salary
<input type="checkbox"/>	1000	Moshiur Rahman	35000.00
<input type="checkbox"/>	1001	Anisur Rahman	45000.00
<input type="checkbox"/>	1002	Moshiur Rahman	NULL
<input type="checkbox"/>	1003	Dev Ed	NULL

```
1 INSERT INTO Teacher(Name, Salary)
2 VALUES
3 ('Al Amin Nahid', 17000.00),
4 ('Abu Taleb', 23000.00),
5 ('Mahbub Alam', 20000.00)
6 ;
```

	ID	Name	Salary
<input type="checkbox"/>	1000	Moshiur Rahman	35000.00
<input type="checkbox"/>	1001	Anisur Rahman	45000.00
<input type="checkbox"/>	1002	Moshiur Rahman	NULL
<input type="checkbox"/>	1003	Dev Ed	NULL
<input type="checkbox"/>	1004	Al Amin Nahid	17000.00
<input type="checkbox"/>	1005	Abu Taleb	23000.00
<input type="checkbox"/>	1006	Mahbub Alam	20000.00

Update

Syntax →

```
UPDATE table_name  
SET column1 = value1, column2 = value2  
WHERE CONDITION
```

	ID	Name	Salary
<input type="checkbox"/>	1000	Moshiur Rahman	35000.00
<input type="checkbox"/>	1001	Anisur Rahman	45000.00
<input type="checkbox"/>	1002	Moshiur Rahman	NULL
<input type="checkbox"/>	1003	Dev Ed	NULL
<input type="checkbox"/>	1004	Al Amin Nahid	17000.00
<input type="checkbox"/>	1005	Abu Taleb	23000.00
<input type="checkbox"/>	1006	Mahbub Alam	20000.00

```
1 UPDATE Teacher  
2 SET Salary = 39000.00  
3 WHERE ID = 1002;
```

	ID	Name	Salary
<input type="checkbox"/>	1000	Moshiur Rahman	35000.00
<input type="checkbox"/>	1001	Anisur Rahman	45000.00
<input type="checkbox"/>	1002	Moshiur Rahman	39000.00
<input type="checkbox"/>	1003	Dev Ed	NULL
<input type="checkbox"/>	1004	Al Amin Nahid	17000.00
<input type="checkbox"/>	1005	Abu Taleb	23000.00
<input type="checkbox"/>	1006	Mahbub Alam	20000.00

```
1 UPDATE Teacher  
2 SET Name = 'Jonas'  
3 WHERE ID = 1002;
```

+ Options

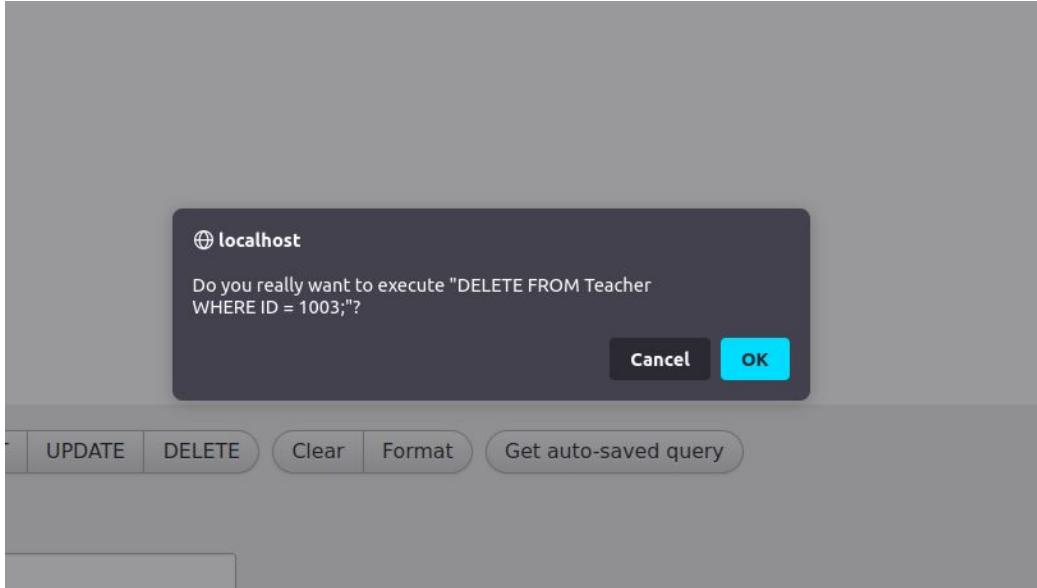
	ID	Name	Salary
<input type="checkbox"/>  Edit  Copy  Delete	1000	Moshiur Rahman	35000.00
<input type="checkbox"/>  Edit  Copy  Delete	1001	Anisur Rahman	45000.00
<input type="checkbox"/>  Edit  Copy  Delete	1002	Jonas	39000.00
<input type="checkbox"/>  Edit  Copy  Delete	1003	Dev Ed	NULL
<input type="checkbox"/>  Edit  Copy  Delete	1004	Al Amin Nahid	17000.00
<input type="checkbox"/>  Edit  Copy  Delete	1005	Abu Taleb	23000.00
<input type="checkbox"/>  Edit  Copy  Delete	1006	Mahbub Alam	20000.00

DELETE

Syntax →

```
DELETE FROM Teacher  
WHERE ID = 1002;
```

```
1 DELETE FROM Teacher  
2 WHERE ID = 1003;
```



A screenshot of a MySQL query editor interface. At the top, there's a toolbar with buttons for UPDATE, DELETE, Clear, Format, and Get auto-saved query. Below the toolbar, a large text area contains the SQL command: "DELETE FROM Teacher WHERE ID = 1003;". A modal dialog box is displayed in the center, titled "localhost". It contains the message: "Do you really want to execute \"DELETE FROM Teacher WHERE ID = 1003;\"?" with "Cancel" and "OK" buttons.

	ID	Name	Salary
<input type="checkbox"/>	1000	Moshiur Rahman	35000.00
<input type="checkbox"/>	1001	Anisur Rahman	45000.00
<input type="checkbox"/>	1002	Jonas	39000.00
<input type="checkbox"/>	1004	Al Amin Nahid	17000.00
<input type="checkbox"/>	1005	Abu Taleb	23000.00
<input type="checkbox"/>	1006	Mahbub Alam	20000.00

```
1 UPDATE Teacher  
2 SET Salary = Salary + 39000.00  
3 WHERE Salary > 20000.00;
```

	<input type="button" value="←"/>	<input type="button" value="→"/>	<input type="button" value="▼"/>	ID	Name	Salary
	<input type="checkbox"/>	 Edit	 Copy	 Delete	1000	Moshiur Rahman 74000.00
	<input type="checkbox"/>	 Edit	 Copy	 Delete	1001	Anisur Rahman 84000.00
	<input type="checkbox"/>	 Edit	 Copy	 Delete	1002	Jonas 78000.00
	<input type="checkbox"/>	 Edit	 Copy	 Delete	1004	Al Amin Nahid 17000.00
	<input type="checkbox"/>	 Edit	 Copy	 Delete	1005	Abu Taleb 62000.00
	<input type="checkbox"/>	 Edit	 Copy	 Delete	1006	Mahbub Alam 20000.00

UPPER()

```
1 SELECT UPPER('maruf');
```

+ Options

Upper('maruf')

MARUF

```
1 SELECT UPPER(City)
2 FROM student_details;
```

+ Options

UPPER(City)

DHAKA

CUMILLA

SYLHET

BARISHAL

RANGPUR

NOWAKHALI

CHITTAGONG

SYLHET

LOWER()

```
1 SELECT LOWER('MARUF');
```

+ Options

LOWER('MARUF')

maruf

```
1 SELECT LOWER(City)
2 FROM student_details;
```

+ Options

LOWER(City)

dhaka

cumilla

sylhet

barishal

rangpur

nowakhali

chittagong

sylhet

CONCAT()

```
1 SELECT CONCAT('Welcome to MySQL ', 'Database');
```

+ Options

CONCAT('Welcome to MySQL ', 'Database')

Welcome to MySQL Database

```
1 SELECT CONCAT(  
2     Name, ' is ', Age, 'years old. He lives in ', City  
3 )  
4 FROM student_details;
```

+ Options

CONCAT(Name, ' is ', Age, 'years old. He lives in ', City)

Maruf is 21years old. He lives in Dhaka

Arif is 17years old. He lives in Cumilla

Mahabuba is 18years old. He lives in Sylhet

Shimul is 16years old. He lives in Barishal

Sharmin is 18years old. He lives in Rangpur

Nayan is 19years old. He lives in Nowakhali

Murad is 20years old. He lives in Chittagong

Anika is 21years old. He lives in Sylhet

```
1 SELECT CONCAT(  
2     Name, ' is ', Age, 'years old. He lives in ', City  
3 ) AS Student  
4 FROM student_details;
```

+ Options

Student

Maruf is 21years old. He lives in Dhaka

Arif is 17years old. He lives in Cumilla

Mahabuba is 18years old. He lives in Sylhet

Shimul is 16years old. He lives in Barishal

Sharmin is 18years old. He lives in Rangpur

Nayan is 19years old. He lives in Nowakhali

Murad is 20years old. He lives in Chittagong

Anika is 21years old. He lives in Sylhet

Greatest()

```
1 SELECT GREATEST(1,2,3,4,5,6,-1, 0, -5, -10, 34,56) AS 'Largest Number';
```

+ Options

Largest Number

56

Least()

```
1 SELECT LEAST(1,2,3,4,5,6,-1, 0, -5, -10, 34,56) AS 'Lowest Number';
```

+ Options

Lowest Number

-10

Power()

```
1 SELECT POW(2,3) AS Power;
```

+ Options

Power

8

LOG()

```
1 | SELECT LOG(2);
```

+ Options

LOG(2)

0.6931471805599453

```
1 | SELECT LOG10(2);
```

+ Options

LOG10(2)

0.3010299956639812

TRUNCATE()

```
1 | SELECT TRUNCATE(3123.3472947219423, 3);
```

+ Options

TRUNCATE(3123.3472947219423, 3)

3123.347

```
1 SELECT TRUNCATE(LOG(2), 3);
```

+ Options

TRUNCATE(LOG(2), 3)

0.693

Rand()

```
1 SELECT RAND();
```

+ Options

RAND()

0.8210786029892775

+ Options

RAND()

0.5234223702209279

+ Options

RAND()

0.6272086413830599

+ Options

RAND()

0.30196899948825034

+ Options

RAND()

0.23955296374815793

EXP() exponential

```
1 SELECT EXP(3);
```

+ Options

EXP(3)

20.085536923187668

```
1 SELECT TRUNCATE(EXP(3),2);
```

+ Options

TRUNCATE(EXP(3),2)

20.08

Group function / Aggregate function → AVG, COUNT, MAX, MIN, SUM

```
1 SELECT AVG(Salary)
2 FROM Teacher;
```

+ Options

AVG(Salary)

55833.333333

```
1 SELECT COUNT(*)
2 FROM student_details;
```

+ Options

COUNT(*)

8

```
1 SELECT MAX(GPA)
2 FROM student_details
```

+ Options

MAX(GPA)

4.89

```
1 SELECT MIN(GPA)
2 FROM student_details
```

+ Options

MIN(GPA)

4.43

```
1 SELECT SUM(Salary)
2 FROM Teacher;
```

+ Options

SUM(Salary)

335000.00

```
1 SELECT MIN(Salary), MAX(Salary), SUM(Salary), AVG(Salary)
2 FROM Teacher;
```

+ Options

MIN(Salary)

17000.00

MAX(Salary)

84000.00

SUM(Salary)

335000.00

AVG(Salary)

55833.333333

```
1 SELECT MIN(GPA)
2 FROM student_details
3 WHERE Gender = 'Male';
```

+ Options

MIN(GPA)

4.52

```
1 SELECT Roll, Name, Age, City, MIN(GPA)
2 FROM student_details
3 WHERE Gender = 'Male';
```

+ Options

← →

▼

Roll	Name	Age	City	MIN(GPA)
------	------	-----	------	----------



Edit



Copy



Delete

101

Maruf

21

Dhaka

4.52

Sub Query

```
1 SELECT AVG(Salary)  
2 FROM Teacher;
```

+ Options
AVG(Salary)
55833.333333

```
1 SELECT * FROM Teacher  
2 WHERE Salary > 55833.333333;
```

	<input type="button" value="←"/> <input type="button" value="→"/>	<input type="button" value="↶"/> <input type="button" value="↷"/>	ID	Name	Salary
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	1000	Moshiur Rahman	74000.00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	1001	Anisur Rahman	84000.00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	1002	Jonas	78000.00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	1005	Abu Taleb	62000.00

```
1 SELECT * FROM Teacher  
2 WHERE Salary > (SELECT AVG(Salary) FROM Teacher);
```

+ Options

		ID	Name	Salary
<input type="checkbox"/>	Edit Copy Delete	1000	Moshiur Rahman	74000.00
<input type="checkbox"/>	Edit Copy Delete	1001	Anisur Rahman	84000.00
<input type="checkbox"/>	Edit Copy Delete	1002	Jonas	78000.00
<input type="checkbox"/>	Edit Copy Delete	1005	Abu Taleb	62000.00

ALTER TABLE

- ALTER TABLE command এর সাহায্যে
 1. টেবিলে নতুন কলাম যুক্ত করা যায় ।
 2. কোন কলামের নাম পরিবর্তন করা যায়।
 3. কোন কলাম delete করা যায় ।

```
1 ALTER TABLE Teacher  
2 ADD Department varchar(20);
```

+ Options

← →

▼

	ID	Name	Salary	Department
<input type="checkbox"/>	1000	Moshiur Rahman	74000.00	NULL
<input type="checkbox"/>	1001	Anisur Rahman	84000.00	NULL
<input type="checkbox"/>	1002	Jonas	78000.00	NULL
<input type="checkbox"/>	1004	Al Amin Nahid	17000.00	NULL
<input type="checkbox"/>	1005	Abu Taleb	62000.00	NULL
<input type="checkbox"/>	1006	Mahbub Alam	20000.00	NULL

↑

Check all

With selected:

 Edit

 Copy

 Delete

 Export

The End