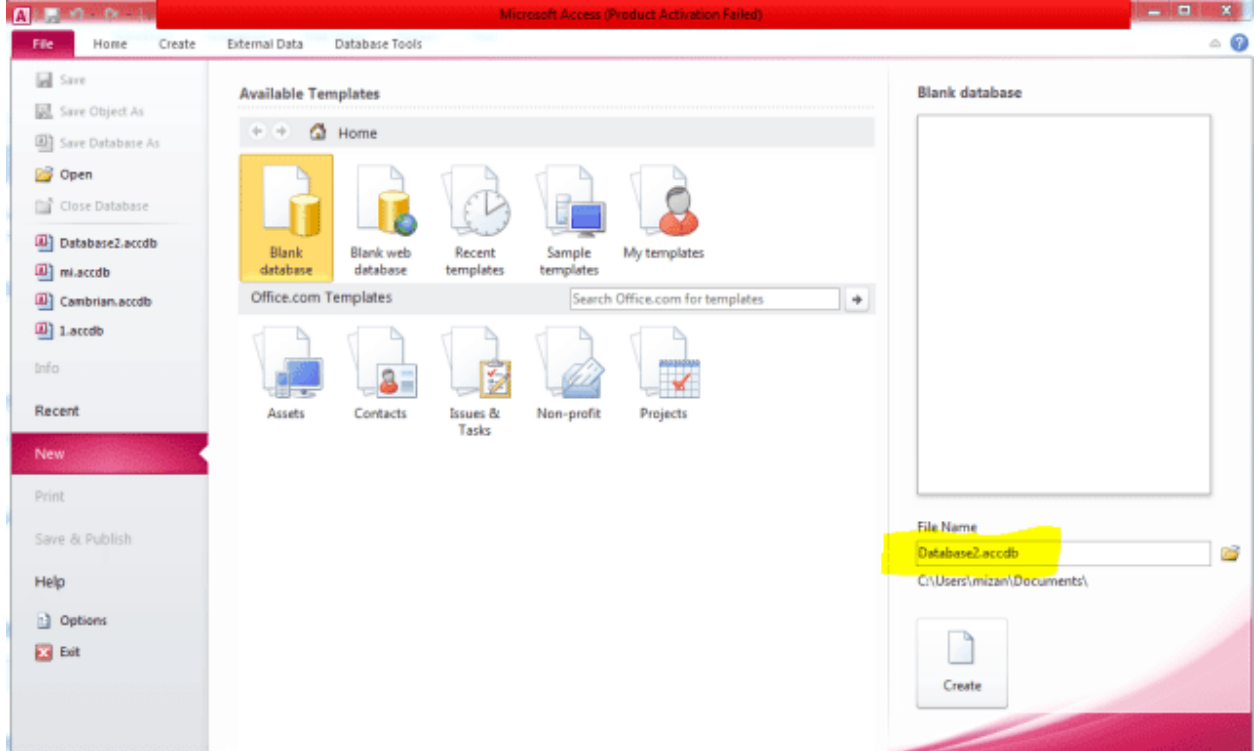


# ষষ্ঠ অধ্যায় পাঠ-৩: ডেটাবেজ ও টেবিল তৈরি এবং ফিল্ডের ডেটা টাইপ।

**ডেটাবেজ তৈরি:** Database তৈরি করার জন্য DBMS হিসেবে Microsoft Access 2010 এর ব্যবহার দেখানো হল। প্রথমেই Microsoft Access 2010 অ্যাপ্লিকেশনটি ইন্সটল দিতে হবে। তারপর প্রোগ্রামটি রান করলে নিচের মত করে চিত্র প্রদর্শিত হবে।



চিত্রের ডান পাশে নিচে File Name থেকে ডেটাবেজের নাম এবং ডেটাবেজের লোকেশন পরিবর্তন করা যায়। অবশেষে create button এ ক্লিক করলে ডেটাবেজ তৈরি হবে।

**টেবিল তৈরি:** ডেটাবেজ হলো পরস্পর সম্পর্কযুক্ত একাধিক টেবিলের সমন্বয়ে গঠিত। প্রত্যেকটি টেবিল আবার কতকগুলো রেকর্ড নিয়ে গঠিত। পরস্পর সম্পর্কযুক্ত কতকগুলো ফিল্ড মিলে গঠিত হয় রেকর্ড। সুতরাং ফিল্ড হচ্ছে ডেটাবেজের ভিত্তি। কোনো টেবিল তৈরি করার পূর্বে টেবিলের প্রত্যেকটি রেকর্ডে কি কি ফিল্ড থাকবে তা নির্দিষ্ট করতে হয়। কোনো ডেটাবেজে কি কি ফিল্ড থাকবে তা নির্ভর করবে ডেটাবেজের উদ্দেশ্য বা ডেটাবেজে কী ধরনের ডেটা থাকবে তার উপর। আবার প্রত্যেকটি ফিল্ডে কী ধরনের ডেটা থাকবে অর্থাৎ ডেটা টাইপ কি হবে তা নির্ধারণ করতে হয়।

ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

## ডেটাবেজ টেবিলের ফিল্ডের ডেটা টাইপ সমূহ:

**Short Text:** বেশিরভাগ ডেটাবেজে ব্যবহৃত প্রধান Data type হলো Text। Short Text ফিল্ডে অক্ষর, সংখ্যা, চিহ্ন ইত্যাদি ব্যবহার করা যায়। সাধারণত এ ফিল্ডে সর্বোচ্চ ২৫৫টি বর্ণ/ অক্ষর/চিহ্ন এককভাবে বা সম্মিলিতভাবে ব্যবহার করা যায়। তবে, সংখ্যা ব্যবহার করলেও এ ডেটার উপর গাণিতিক কাজ করা যায় না।

**Long Text:** Long Text ফিল্ডে অক্ষর, সংখ্যা, চিহ্ন ইত্যাদি ব্যবহার করা যায়। সাধারণত এ ফিল্ডে ৬৩৯৯৯ সংখক বর্ণ/ অক্ষর/চিহ্ন এককভাবে বা সম্মিলিতভাবে ব্যবহার করা যায়। তবে, সংখ্যা ব্যবহার করলেও এ ডেটার উপর গাণিতিক কাজ করা যায় না।

**Number:** যে ফিল্ডে গাণিতিক ডেটা ব্যবহার করা হয়, সেই ফিল্ডকে প্রকাশ করার জন্য নাম্বার ব্যবহৃত হয়। নাম্বার ফিল্ডে যোগ বা বিয়োগ চিহ্নসহ/ছাড়া পূর্ণসংখ্যা ও ভগ্নাংশ মিলিয়ে প্রয়োজনীয় সংখ্যা ব্যবহার করা যায়। এ ফিল্ডের ডেটার উপর গাণিতিক অপারেশন (যোগ, বিয়োগ, গুণ ও ভাগ) করা যায়।

**AutoNumber:** এটি একটি নাম্বার ডেটা টাইপ। এটি সিরিজ জাতীয় বা ধারাবাহিক ডেটার ক্ষেত্রে ব্যবহৃত হয়। এ ডেটা টাইপের সুবিধা হচ্ছে এতে ডেটা এন্ট্রি করতে হয় না। স্বয়ংক্রিয়ভাবে ডেটা এন্ট্রি হয়।

**Yes/No:** যুক্তিনির্ভর ফিল্ডের ডেটা টাইপ প্রকাশ করার জন্য Yes/No ব্যবহৃত হয়। কোনো ফিল্ডের মান 'হ্যাঁ' অথবা 'না' এ দুটি তথ্য এ ফিল্ডে সংরক্ষণ করা যায়। এই ফিল্ডের জন্য মেমোরিতে ১ বিট জায়গা প্রয়োজন।

**Date/Time:** এ ফিল্ডটি তারিখ বা সময়ের জন্য ব্যবহার করা হয়। ১০০ থেকে ৯৯৯৯ বছরের তারিখ ও সময়ের জন্য এ ফিল্ড ব্যবহৃত হয়। এ ফিল্ডের জন্য মেমোরিতে ৮ বাইট জায়গা প্রয়োজন। তারিখ ও সময় বিভিন্ন ফরমেটে হতে পারে।

**Memo:** Memo, Text এর পরিপূরক হিসেবে ব্যবহৃত হয়। সাধারণত বর্ণনামূলক লেখা বা বর্ণনার জন্য এ ফিল্ড ব্যবহার করা হয়। এ ফিল্ডের ধারণ ক্ষমতা কম্পিউটার ডিস্কের ধারণ ক্ষমতার উপর নির্ভর করে। সাধারণত এ ফিল্ডে ৬৫,৫৩৬টি ক্যারেক্টর লেখা যায়। সাধারণত Remark, Address ফিল্ডে এ ডেটা টাইপ ব্যবহার করা হয়।

**Currency:** মুদ্রা বা টাকার অঙ্ক ইনপুট করার জন্য \$ ব্যবহার করা হয়। শুধুমাত্র মুদ্রা বা টাকা সংক্রান্ত ডেটা এন্ট্রি করার জন্য Currency টাইপ সিলেক্ট করতে হয়। এ ফিল্ডের ডেটার উপর গাণিতিক অপারেশন সম্পূর্ণ প্রযোজ্য। এ ফিল্ডের জন্য মেমোরিতে ৮ বাইট জায়গা প্রয়োজন।

**OLE(Object Linking Embedding) Object:** যেসব তথ্য ডেটাবেজ নয় এমন সফটওয়্যারে আছে এবং লিংক এর মাধ্যমে স্বয়ংক্রিয়ভাবে ডেটাবেজে নেয়ার ক্ষেত্রে এ ডেটা টাইপ ব্যবহার করা হয়। যেমন- মাইক্রোসফট এক্সেল, পাওয়ার পয়েন্ট ইত্যাদি প্রোগ্রাম হতে শব্দ, ছবি, টেক্সট, গ্রাফ ইত্যাদি ডেটাবেজের কোন ফিল্ডে নেয়ার জন্য এ ডেটা টাইপ ব্যবহার করা হয়।

**Hyperlink:** সাধারণত ডেটাবেজ প্রোগ্রামের সাথে ওয়েব পেজের কোনো ফাইল কিংবা অন্য কোনো ব্যবহারিক প্রোগ্রামের ফাইল লিংক করার জন্য এ ডেটা টাইপ ব্যবহার করা হয়।

**Look up wizard:** সরাসরি কোনো ডেটা এন্ট্রি না করে কোনো লিস্ট বা টেবিল থেকে ডেটা নির্বাচন করে ডেটা ইনপুট করার জন্য এ ডেটা টাইপ ব্যবহার করা হয়।

ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**ষষ্ঠ অধ্যায় পাঠ-৪: ডেটাবেজ কুয়েরি, বিভিন্ন কুয়েরি ভাষা এবং অপারেটর সমূহ।**

**কুয়েরিঃ** ডেটাবেজে সংরক্ষিত অসংখ্য তথ্য থেকে নির্দিষ্ট কোনো শর্ত সাপেক্ষে তথ্য খুঁজে বের করাকে বলা হয় কুয়েরি। কুয়েরির সাহায্যে নির্দিষ্ট ফিল্ডের ডেটা, নির্দিষ্ট গ্রুপের ডেটা নির্দিষ্ট শর্ত সাপেক্ষে প্রদর্শন করা যায়।

**বিভিন্ন প্রকার কুয়েরিঃ**

- **সিলেক্ট কুয়েরি (Select Query):** কোনো ডেটাবেজ টেবিলের ফিল্ড বা কলাম নির্বাচন করে যে কুয়েরি করা হয় তাকে সিলেক্ট কুয়েরি বলা হয়।
- **প্যারামিটার কুয়েরি (Parameter Query):** ডায়ালগ বক্সের তথ্য পূরণ করে যে কুয়েরি করা হয় তাকে প্যারামিটার কুয়েরি বলে।
- **ক্রসট্যাব কুয়েরি (Cross tab Query):** শর্তারোপ করে কুয়েরিকৃত ফলাফল সামারি আকারে প্রদর্শনের জন্য যে কুয়েরি করা হয় তাকে, ক্রসট্যাব কুয়েরি বলে।
- **অ্যাকশন কুয়েরি (Action Query):** কোনো কুয়েরি যখন ডেটাবেজের ডেটার মানের পরিবর্তন করে তাকে অ্যাকশন কুয়েরি বলে। যেমন-
  - Append Query- নতুন রেকর্ড যুক্ত করার জন্য ব্যবহার করা হয়।
  - Update Query -কোনো ডেটার পরিবর্তনের জন্য ব্যবহার করা হয়।
  - Delete Query- টেবিল থেকে ডেটা মুছে ফেলার জন্য ব্যবহার করা হয়।
  - Make Table Query- কুয়েরিকৃত ফলাফল দিয়ে নতুন টেবিল তৈরির জন্য ব্যবহার করা হয়।

**কুয়েরি ভাষাঃ** যে ভাষার সাহায্যে কুয়েরি করা হয় তাকে কুয়েরি ভাষা বলে। ডেটা ম্যানিপুলেশনের উপর ভিত্তি করে তিন ধরনের কুয়েরি ভাষা আছে। যেমন-

- 1. QUEL (Query Language)
- 2. QBE (Query By Example)
- 3. SQL (Structured Query Language)

**QUEL:** QUEL এর পূর্ণ রূপ হলো Query Language। এটি INGRES এর জন্য ডেটা ডেফিনিশন এবং ডেটা ম্যানিপুলেশন। INGRES এর পূর্ণ রূপ হলো Interactive Graphics and Retrieval System। INGRES একটি রিলেশনাল ডেটাবেজ ম্যানেজমেন্ট সিস্টেম যা মাইকেল স্টোনব্রাকারের(Michael Stonebraker) তৈরি করেন। QUEL রিলেশনাল বীজগণিত অপারেশনগুলি যেমন-ইন্টারসেকশন, বিয়োগ বা ইউনিয়ন সমর্থন করে না। এটি টাপল ক্যালকুলাসের উপর ভিত্তি করে তৈরি এবং এটি নেস্টেড সাব কুয়েরিগুলিকে সমর্থন করে না।

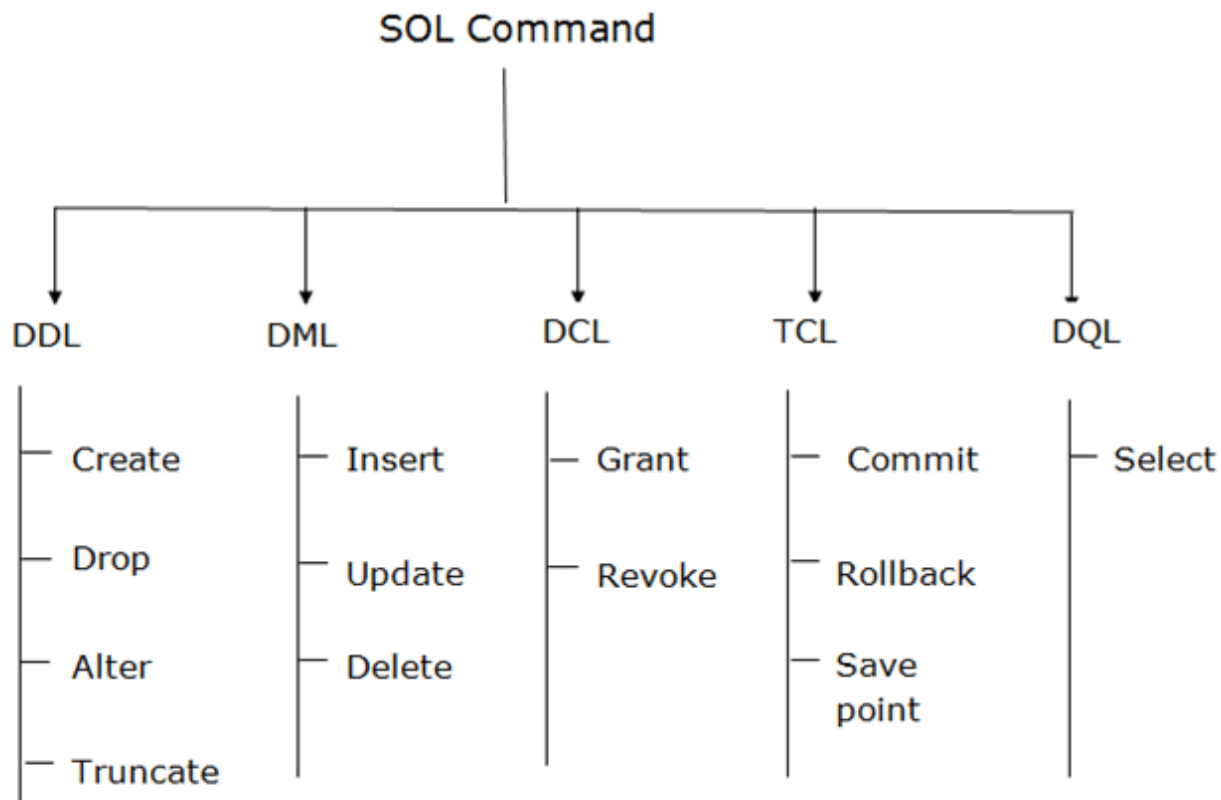
**QBE:** QBE এর পূর্ণ রূপ হলো Query by example। এটি একটি কুয়েরি ভাষা যা রিলেশনাল ডেটাবেজে ব্যবহৃত হয়। এটি টেবিল থেকে তথ্য অনুসন্ধানের জন্য ব্যবহারকারীদের একটি সাধারণ ইউজার ইন্টারফেস সরবরাহ করে যেখানে ব্যবহারকারী তার যে তথ্যটি অ্যাক্সেস করতে চায় তার একটি উদাহরণ ইনপুট করতে সক্ষম হবে। SQL এর উন্নয়নের সমান্তরালে 1970-এর দশকে IBM এ Moshe Zloof কর্তৃক QBE তৈরি করা হয়েছিল। এটি একটি গ্রাফিক্যাল কুয়েরি ভাষা যেখানে ব্যবহারকারীরা শর্ত এবং উদাহরণ উপাদানগুলির মতো কমান্ডগুলি টেবিলে ইনপুট করতে পারে। ব্যাকগ্রাউন্ডে ব্যবহারকারীর কুয়েরিটি ডেটাবেজ ম্যানিপুলেশন ভাষায় (যেমন SQL) রূপান্তরিত হয় এবং এই SQL ব্যাকগ্রাউন্ডে কার্যকর হবে।

**SQL(Structured Query Language):** SQL এর পূর্ণ রূপ হলো Structured Query Language। SQL একটি non-procedural বা Functional Language। কারণ SQL এ যে তথ্যাবলি দরকার কেবল তা বলে দিলেই হয়, কীভাবে কুয়েরি করা যাবে তা বলার দরকার হয় না। SQL একটি শক্তিশালী ডেটা ডেফিনিশন ল্যাঙ্গুয়েজ অর্থাৎ SQL ব্যবহার করে ডেটাবেজ ম্যানেজমেন্ট সিস্টেমে ডেটাবেজ ফাইল তৈরি, ডেটাবেজ ফাইল পরিবর্তন, ডেটাবেজ ফাইল ডিলিট, ডেটাবেজ অবজেক্ট(টেবিল, ভিউ, ইন্ডেক্স ইত্যাদি) তৈরি, পরিবর্তন এবং ডিলিট করা যায় এবং SQL একটি শক্তিশালী ডেটা মেনিপুলেশন ল্যাঙ্গুয়েজ অর্থাৎ SQL ব্যবহার করে ডেটাবেজ টেবিলে ডেটা ইনসার্ট, আপডেট ও ডিলিট করা যায়। তাছাড়া ইউজার একসেস নিয়ন্ত্রণ করা যায়। এ কারণে SQL কে ডেটাবেজের জন্য একটি শক্তিশালী হাতিয়ার বলা হয়।

SQL বিভিন্ন স্টেটমেন্ট এর সমন্বয়ে গঠিত। যেমন-

- 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)

বিভিন্ন SQL স্টেটমেন্ট বা কমান্ডসমূহঃ



**Data Definition Language (DDL):** DDL এর সাহায্যে টেবিলের গঠন পরিবর্তন যেমন- টেবিল তৈরি, টেবিল আপডেট, টেবিল মুছে ফেলা ইত্যাদি পরিবর্তনগুলি করা হয়। সকল DDL কমান্ডগুলি auto-committed। এর মানে এটি ডেটাবেজে স্থায়ীভাবে সমস্ত পরিবর্তন সংরক্ষণ করে।

DDL কমান্ডসমূহ:

**CREATE-** ডেটাবেজে নতুন টেবিল তৈরি করতে ব্যবহৃত হয়। নিচের সিনট্যাক্সটি লক্ষ্য কর-

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

**ALTER-** ডেটাবেজের স্ট্রাকচার পরিবর্তন করতে ব্যবহৃত হয়। যেমন- টেবিলে নতুন কোন ফিল্ড যোগ করা বা ডিলিট করা ইত্যাদি। নিচের সিনট্যাক্সটি লক্ষ্য কর-

`ALTER TABLE table_name ADD column_name COLUMN-definition;`

**DROP-** ডেটাবেজের টেবিল মুছে ফেলতে ব্যবহৃত হয়। নিচের সিনট্যাক্সটি লক্ষ্য কর-

`DROP TABLE table_name;`

**TRUNCATE-** টেবিলের সকল রেকর্ড মুছে ফেলতে এবং টেবিলের স্পেস খালি করতে ব্যবহৃত হয়। নিচের সিনট্যাক্সটি লক্ষ্য কর-

TRUNCATE TABLE table\_name;

**RENAME-** টেবিলের নাম পরিবর্তন করতে ব্যবহৃত হয়।

ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**Data Manipulation Language (DML):** DML কমান্ডসমূহ টেবিলে সংরক্ষিত তথ্য ম্যানিপুলেট করার জন্য ব্যবহৃত হয়। DML কমান্ডসমূহ auto-committed নয়। এর মানে হল পরিবর্তনগুলি ডেটাবেজের জন্য স্থায়ী নয়, তারা আবার পূর্বাবস্থায় আসতে পারে।

DML কমান্ডসমূহ:

**INSERT** – টেবিলের সারিতে ডেটা ইনসার্ট করার জন্য ব্যবহৃত হয়। নিচের সিনট্যাক্সটি লক্ষ্য কর-

INSERT INTO

table\_name(col1, col2, col3,... col N) VALUES (value1, value2, value3, .... valueN);

**UPDATE-** টেবিলের কলামের মান পরিবর্তন করতে ব্যবহৃত হয়। নিচের সিনট্যাক্সটি লক্ষ্য কর-

UPDATE table\_name SET [column\_name1= value1,...column\_nameN = valueN] [WHERE CONDITION]

**DELETE-** টেবিলের এক বা একাধিক সারি বা রেকর্ড ডিলিট করতে ব্যবহৃত হয়। নিচের সিনট্যাক্সটি লক্ষ্য কর-

DELETE FROM table\_name [WHERE condition];

**Transaction Control Language(TCL):** TCL কমান্ডগুলো ডেটাবেজের উপর অন্যান্য কমান্ডগুলোর প্রভাব চেক করে। TCL কমান্ডগুলো কেবলমাত্র ইনসার্ট, ডিলিট এবং আপডেটের মতো DML কমান্ডগুলোর সাথে ব্যবহার হতে পারে। এই অপারেশনগুলো স্বয়ংক্রিয়ভাবে ডেটাবেজে committed হয়, তাই তারা টেবিল তৈরি করার সময় বা তাদের ড্রপ করার সময় ব্যবহার করা যাবে না।

TCL কমান্ডসমূহ:

**COMMIT-** ডেটাবেজের সকল ট্রানজেকশন সংরক্ষণ করতে ব্যবহৃত হয়।

COMMIT;

**ROLLBACK-** যেসকল ট্রানজেকশন এখনো ডেটাবেজে সংরক্ষণ হয় নাই তাদেরকে undo করতে এই কমান্ড ব্যবহৃত হয়।

ROLLBACK;

**SAVEPOINT-** সমস্ত ট্রানজেকশনটি roll back না করে একটি নির্দিষ্ট পয়েন্টে ট্রানজেকশনটি roll back করার জন্য ব্যবহৃত হয়।

SAVEPOINT SAVEPOINT\_NAME;

**Data Control Language (DCL):** DCL কমান্ডসমূহ ডেটাবেজের যেকোন ব্যবহারকারীকে অথরিটির এক্সেস দেওয়া এবং তা ফিরিয়ে নিতে ব্যবহৃত হয়।

DCL কমান্ডসমূহ:

**GRANT-** ডেটাবেজে ইউজার এক্সেস সুবিধা দিতে ব্যবহৃত হয়।

GRANT SELECT, UPDATE ON MY\_TABLE TO SOME\_USER, ANOTHER\_USER;

**REVOKE-** ডেটাবেজে ইউজার এক্সেস সুবিধা বাতিল করতে ব্যবহৃত হয়।

REVOKE SELECT, UPDATE ON MY\_TABLE FROM USER1, USER2;

**Data query language (DQL):** DQL কমান্ডটি ডেটাবেজ টেবিল থেকে শর্ত সাপেক্ষে ডেটা আনতে ব্যবহৃত হয়।

**SELECT-** শর্ত সাপেক্ষে ডেটাবেজ টেবিলের অ্যাট্রিবিউট বা কলাম সিলেক্ট করতে ব্যবহৃত হয়। নিচের সিনট্যাক্সটি লক্ষ্য কর-

SELECT column\_name—/\* FROM table\_name WHERE conditions;

**ডেটাবেজের অপারেটর সমূহঃ** ডেটাবেজের বিভিন্ন SQL কমান্ড লেখার জন্য বিভিন্ন অপারেটর প্রয়োজন হয়। ডেটাবেজের বিভিন্ন অপারেটর সমূহ নিচে আলোচনা করা হল-

- ১। Arithmetic অপারেটর
- ২। Comparison অপারেটর
- ৩। Concatenation অপারেটর
- ৪। Logical অপারেটর
- ৫। Special অপারেটর



**Arithmetic অপারেটরঃ** নিউমেরিক এক্সপ্রেশন তৈরি করতে Arithmetic অপারেটর ব্যবহার করা হয়। যেমন – যোগ, বিয়োগ, গুণ,ভাগ ইত্যাদির জন্য ব্যবহৃত হয়।

অপারেটর	বর্ণনা
+	যোগ চিহ্ন।
-	বিয়োগ চিহ্ন।
*	গুণন চিহ্ন।
/	ভাগ চিহ্ন।
\	পূর্ণ সংখ্যায় ভাগফল নির্ণয়ের জন্য।
Mod	২ টি সংখ্যাকে ভাগ করে ভাগশেষ প্রকাশ করে।
()	গ্রুপ এক্সপ্রেশনের জন্য ব্যবহৃত হয়।
^	এক্সপোনেনশিয়াল।

ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**Comparison অপারেটরঃ** দুটি এক্সপ্রেশনের মধ্যে তুলনা করতে Comparison অপারেটর ব্যবহৃত হয়। একে রিলেশনাল অপারেটরও বলা হয়।

অপারেটর	বর্ণনা
=	সমান চিহ্ন।
>	অপারেটরের বাম অংশ ডান অংশের চেয়ে বড়।
>=	অপারেটরের বাম অংশ ডান অংশের চেয়ে বড় অথবা সমান।
<	অপারেটরের বাম অংশ ডান অংশের চেয়ে ছোট।
<=	অপারেটরের বাম অংশ ডান অংশের চেয়ে ছোট অথবা সমান।
<>	সমান নয়/ অসমান।
Between .. And	রেঞ্জ বুঝানোর জন্য ব্যবহৃত হয়।



## ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**Concatenation অপারেটরঃ** দুটি স্ট্রিংকে যোগ করার জন্য Concatenation অপারেটর ব্যবহার করা হয়। & চিহ্ন হচ্ছে

Concatenation অপারেটর। এছাড়া + চিহ্ন দ্বারাও দুটি স্ট্রিংকে যোগ করা যায়।

- যেমন- string\_1="Con" এবং string\_2="catenation"
- result = string\_1 & string\_2 অথবা result = string\_1 + string\_2

**Logical অপারেটরঃ** দুটি এক্সপ্রেশনকে একত্র করে এবং এক্সপ্রেশনটি সত্য বা মিথ্যা মূল্যায়ন করে।

অপারেটর	বর্ণনা
And	উভয় শর্ত সত্য হতে হবে।
Or	দুটি শর্তের কমপক্ষে একটি শর্ত সত্য হতে হবে।
Not	এক্সপ্রেশনকে মূল্যায়ন করলে শর্তাবলি অবশ্যই মিথ্যা হতে হবে।
Eqn	দুটি এক্সপ্রেশনকে Bitwise তুলনা করে। এটি প্রোগ্রামিং এ ব্যবহৃত হয়।
Imp	দুটি নিউমেরিক এক্সপ্রেশনের মধ্যে লজিক্যাল Implecation সম্পাদন করে।
()	গ্রুপ এক্সপ্রেশনের জন্য।

ষষ্ঠ অধ্যায় পাঠ-৫ ডেটাবেজের বিভিন্ন SQL কমান্ড।

একটি ডেটাবেজ তৈরির SQL কমান্ডের সিনট্যাক্স:

```
1 CREATE DATABASE database_name;
```

db\_sagc নামে একটি ডেটাবেজ তৈরির SQL কমান্ড:

```
1 CREATE DATABASE db_sagc;
```

টেবিল তৈরির SQL কমান্ডের সিনট্যাক্স:

```
SQL command for creating database table
1 CREATE TABLE table_name
2 (
3   column_name_1 data_type(size),
4   column_name_2 data_type(size),
5   .....
6   column_name_n data_type(size)
7 );
```

## student\_info

Id	Name	Section	GPA	City

Smart Learning

student\_info নামের উপরের টেবিলটি তৈরির SQL কমান্ড:

```
1 CREATE TABLE student_info
2 (
3   Id number PRIMARY KEY,
4   Name text(20),
5   Section text(5),
6   GPA number,
7   City text(20)
8 );
```

টেবিলে নতুন ফিল্ড যোগ করার SQL কমান্ডের সিনট্যাক্স:

```
1 ALTER TABLE table_name ADD
2 (
3   new_column_name_1 data_type(size),
4   new_column_name_2 data_type(size)
5 );
```

## student\_info

Id	Name	Section	GPA	City

student\_info নামের উপরের টেবিলে Contact নামে নতুন একটি ফিল্ড যোগ করার SQL কমান্ড:

```
1 ALTER TABLE student_info ADD
2 (
3   Contact text(25)
4 );
```

SQL কমান্ডটি রান করলে টেবিলটির গঠন নিম্নরূপ হবে-

## student\_info

Id	Name	Section	GPA	City	Contact

টেবিল থেকে একটি ফিল্ড মুছে ফেলার SQL কমান্ডের সিনট্যাক্স:

```
1 ALTER TABLE table_name DROP column_name;
```

### DML(Data Manipulation Language) এর ব্যবহারঃ

কোন টেবিলে নতুন রেকর্ড যোগ করার SQL কমান্ডের সিনট্যাক্সঃ

```
SQL command for adding a new record
1 INSERT INTO table_name (column_name_1,column_name_2,..)
2 VALUES(Value_1,Value_2,..);
```

## student\_info

Id	Name	Section	GPA	City

student\_info নামে উপরের টেবিলে নতুন রেকর্ড যোগ করার SQL কমান্ডঃ

```
1 INSERT INTO student_info(Id, Name, Section, GPA, City) VALUES( 1, "Ashek", "A", 5.00, "Dhaka" );
```

নতুন রেকর্ড যোগ করার ফলে টেবিলটি দেখতে নিম্নরূপ হবে-

## student\_info

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka

Smart Lead

## student\_info

Id	Name	Section	GPA	City

student\_info নামের উপরের টেবিল থেকে City ফিল্ড মুছে ফেলার SQL কমান্ড:

```
1 ALTER TABLE student_info DROP City ;
```

SQL কমান্ডটি রান করলে টেবিলটির গঠন নিম্নরূপ হবে-

## student\_info

Id	Name	Section	GPA

ডেটাবেজ থেকে টেবিল মুছে ফেলার SQL কমান্ডের সিনট্যাক্স:

```
1 DROP TABLE table_name;
```

ডেটাবেজ থেকে student\_info নামের টেবিল মুছে ফেলার SQL কমান্ড:

```
1 DROP TABLE student_info;
```

Smart Learner

DML(Data Manipulation Language) এর ব্যবহারঃ

কোন টেবিলে নতুন রেকর্ড যোগ করার SQL কমান্ডের সিনট্যাক্সঃ

```
SQL command for adding a new record
1 INSERT INTO table_name (column_name_1,column_name_2,..)
2 VALUES(Value_1,Value_2,..);
```

## student\_info

Id	Name	Section	GPA	City

student\_info নামে উপরের টেবিলে নতুন রেকর্ড যোগ করার SQL কমান্ডঃ

```
1 INSERT INTO student_info(Id, Name, Section, GPA, City) VALUES( 1, "Ashek", "A", 5.00, "Dhaka" );
```

নতুন রেকর্ড যোগ করার ফলে টেবিলটি দেখতে নিম্নরূপ হবে-

## student\_info

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka

Smart Learning

এক বা একাধিক রেকর্ড আপডেট করার জন্য SQL কমান্ডের সিনট্যাক্স:

```
Update Query
1 UPDATE table_name
2 SET column_name_1=value, column_name_2=value, ...
3 WHERE condition;
```

## student\_info

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka
2	Amirul	B	4.00	Jessore
3	Shanta	A	5.00	Dhaka
4	Sondha	C	4.50	Jamalpur
5	Sajid	A	5.00	Khulna

student\_info নামে উপরের টেবিলের Id = 2 রেকর্ডটির City এর মান Khulna করার জন্য SQL কমান্ড:

```
1 UPDATE student_info SET City="Khulna" WHERE Id = 2;
```

SQL কমান্ডটি রান করলে টেবিলটি দেখতে নিম্নরূপ হবে-

## student\_info

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka
2	Amirul	B	4.00	Khulna
3	Shanta	A	5.00	Dhaka
4	Sondha	C	4.50	Jamalpur
5	Sajid	A	5.00	Khulna

এক বা একাধিক রেকর্ড ডিলিট করার জন্য SQL কমান্ডের সিনট্যাক্স:

```
Delete Query
1 DELETE FROM table_name WHERE condition;
```



## student\_info

Id	Name	Section	GPA	City
2	Amirul	B	4.00	Jessore
3	Shanta	A	5.00	Dhaka
4	Sondha	C	4.50	Jamalpur
5	Sajid	A	5.00	Khulna

একটি টেবিলের সবগুলো রেকর্ড ডিলিট করার জন্য SQL কমান্ডের সিনট্যাক্স:

```
1 DELETE FROM table_name;
```

DQL (Data Query Language ) এর ব্যবহার:

একটি টেবিল থেকে সকল ফিল্ড এবং রেকর্ড সিলেক্ট করে দেখানোর SQL কমান্ডের সিনট্যাক্স:

```
Select Query
1 SELECT * FROM table_name;
```

## student\_info

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka
2	Amirul	B	4.00	Jessore
3	Shanta	A	5.00	Dhaka
4	Sondha	C	4.50	Jamalpur
5	Sajid	A	5.00	Khulna

student\_info নামে উপরের টেবিলের Id =1 রেকর্ডটি ডিলিট করার জন্য SQL কমান্ড:

```
1 DELETE FROM student_info WHERE Id = 1;
```

SQL কমান্ডটি রান করলে টেবিলটি দেখতে নিম্নরূপ হবে-

## student\_info

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka
2	Amirul	B	4.00	Jessore
3	Shanta	A	5.00	Dhaka
4	Sondha	C	4.50	Jamalpur
5	Sajid	A	5.00	Khulna

একটি টেবিল থেকে নির্দিষ্ট কিছু ফিল্ড এবং সকল রেকর্ড সিলেক্ট করে দেখানোর SQL কমান্ডের সিনট্যাক্স:

```
1 SELECT column_name_1,column_name_2,... FROM table_name;
```

## student\_info

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka
2	Amirul	B	4.00	Jessore
3	Shanta	A	5.00	Dhaka
4	Sondha	C	4.50	Jamalpur
5	Sajid	A	5.00	Khulna

student\_info নামে উপরের টেবিলের নির্দিষ্ট কিছু ফিল্ড এবং সকল রেকর্ড সিলেক্ট করে দেখানোর SQL কমান্ড:

```
1 SELECT Name, Section, GPA FROM student_info;
```

SQL কমান্ডটি রান করলে আউটপুট টেবিলটি দেখতে নিম্নরূপ হবে-

Name	Section	GPA
Ashek	A	5.00
Amirul	B	4.00
Shanta	A	5.00
Sondha	C	4.50
Sajid	A	5.00

টেবিল থেকে নির্দিষ্ট শর্তসাপেক্ষে নির্দিষ্ট রেকর্ড দেখানোর SQL কমান্ডের সিনট্যাক্স:

```
1 SELECT */ column_name_1,column_name_2,.. FROM table_name WHERE condition;
```

### student\_info

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka
2	Amirul	B	4.00	Jessore
3	Shanta	A	5.00	Dhaka
4	Sondha	C	4.50	Jamalpur
5	Sajid	A	5.00	Khulna

student\_info নামে উপরের টেবিলে যাদের GPA=5.00 তাদের তথ্য দেখানোর SQL কমান্ড:

```
1 SELECT * FROM student_info WHERE GPA=5.00;
```

SQL কমান্ডটি রান করলে আউটপুট টেবিলটি দেখতে নিম্নরূপ হবে-

Id	Name	Section	GPA	City
1	Ashek	A	5.00	Dhaka
3	Shanta	A	5.00	Dhaka
5	Sajid	A	5.00	Khulna

**ষষ্ঠ অধ্যায় পার্ট-৬: ডেটাবেজ সার্টিং এবং ইনডেক্সিং।**

**ডেটাবেজ সর্টিংঃ** এক বা একাধিক ফিল্ড এর মানের উপর ভিত্তি করে ডেটাবেজের রেকর্ডগুলোকে উর্ধ্বক্রম বা নিম্নক্রমে সাজানোর প্রক্রিয়া হচ্ছে সর্টিং। একটি ডেটা টেবিল সর্টিং করার ফলে নতুন একটি টেবিল তৈরি হয় যেখানে টেবিলের রেকর্ডগুলো সর্টিং করা থাকে। একটি টেবিল সর্টিং করার পর টেবিলে নতুন কোন ডেটা ইনপুট দিলে সর্টেড টেবিলটির কোন পরিবর্তন হয় না। এই ক্ষেত্রে পুনরায় সর্টিং করতে হয়। এছাড়া কোন টেবিলের ডেটা সর্টিং করলে যেহেতু টেবিলের নতুন একটি কপি তৈরি হয় তাই অতিরিক্ত মেমোরির প্রয়োজন হয়। বাস্তবক্ষেত্রে সর্টিং এর চেয়ে ইনডেক্সিং বেশি ব্যবহৃত হয়।

**সর্টিং দু-প্রকার হয়ে থাকে-**

- ১। উচ্চক্রম/উর্ধ্বক্রম/Ascending order
- ২। নিম্নক্রম/অবরোহী/Descending order

টেবিলের রেকর্ডগুলো GPA ফিল্ডের এর ভিত্তিতে নিম্নক্রমে সর্টিং করা হয়েছে-

### **student\_info**

Roll	Name	GPA
1	Ashek	3.50
2	Amirul	5.00
3	Sajid	4.50
4	Sondha	5.00
5	Shanta	4.90

**Before sorting**

Roll	Name	GPA
2	Amirul	5.00
4	Sondha	5.00
5	Shanta	4.90
3	Sajid	4.50
1	Ashek	3.50

**After sorting**

**ডেটাবেজ সর্টিং এর উদ্দেশ্যঃ**

- ১। ডেটাবেজ সর্টিং এর উদ্দেশ্য হলো কোন ডেটাবেজ টেবিল থেকে কুয়েরির মাধ্যমে প্রাপ্ত আউটপুট ডেটাকে সাজিয়ে উপস্থাপন করা।

টেবিল থেকে ডেটা শর্তসাপেক্ষে সিলেক্ট করে কোন এক বা একাধিক ফিল্ডের উপর ভিত্তি করে রেকর্ড গুলোকে উর্ধ্বক্রম অনুসারে সাজিয়ে দেখানোর SQL কমান্ডের সিনট্যাক্সঃ

```
1 SELECT */ column_name_1,column_name_2,.. FROM table_name
2 WHERE condition ORDER BY column_name ASC;
```

## student\_info

Id	Name	Section	GPA	City
2	Ashek	A	5.00	Dhaka
3	Amirul	B	4.00	Jessore
1	Shanta	A	5.00	Dhaka
5	Sondha	C	4.50	Jamalpur
4	Sajid	A	5.00	Khulna

student\_info নামে উপরের টেবিলের ডেটা সিলেক্ট করে Id ফিল্ডের উপর ভিত্তি করে রেকর্ড গুলোকে উর্ধ্বক্রম অনুসারে সাজিয়ে দেখানোর SQL কমান্ডঃ

```
1 SELECT * FROM student_info ORDER BY Id ASC;
```

SQL কমান্ডটি রান করলে আউটপুট টেবিলটি দেখতে নিম্নরূপ হবে-

Id	Name	Section	GPA	City
1	Shanta	A	5.00	Dhaka
2	Ashek	A	5.00	Dhaka
3	Amirul	B	4.00	Jessore
4	Sajid	A	5.00	Khulna
5	Sondha	C	4.50	Jamalpur

টেবিল থেকে ডেটা সিলেক্ট করে অধঃক্রম অনুসারে সাজিয়ে দেখানোর SQL কমান্ডের সিনট্যাক্সঃ

```
1 SELECT */ column_name_1,column_name_2,.. FROM table_name
2 ORDER BY column_name DESC;
```

## student\_info

Id	Name	Section	GPA	City
2	Ashek	A	5.00	Dhaka
3	Amirul	B	4.00	Jessore
1	Shanta	A	5.00	Dhaka
5	Sondha	C	4.50	Jamalpur
4	Sajid	A	5.00	Khulna

student\_info নামে উপরের টেবিলের ডেটা সিলেক্ট করে Id ফিল্ডের উপর ভিত্তি করে রেকর্ড গুলোকে অধঃক্রম অনুসারে সাজিয়ে দেখানোর SQL কমান্ড:

```
1 SELECT * FROM student_info ORDER BY Id DESC;
```

SQL কমান্ডটি রান করলে আউটপুট টেবিলটি দেখতে নিম্নরূপ হবে-

Id	Name	Section	GPA	City
5	Sondha	C	4.50	Jamalpur
4	Sajid	A	5.00	Khulna
3	Amirul	B	4.00	Jessore
2	Ashek	A	5.00	Dhaka
1	Shanta	A	5.00	Dhaka

**ইনডেক্সিংঃ** ইনডেক্সিং হচ্ছে সুসজ্জিতভাবে বা সুবিন্যস্তভাবে তথ্যাবলির সূচি প্রণয়ন করা। ডেটাবেজ থেকে ব্যবহারকারি কোনো ডেটা যাতে দ্রুত খুঁজে বের করতে পারে সেজন্য ডেটাকে একটি বিশেষ অর্ডারে সাজিয়ে ডেটাগুলোর একটা সূচি প্রণয়ন করা হয়। ডেটাবেজ টেবিলের রেকর্ড সমূহকে এরূপ কোনো লজিক্যাল অর্ডারে সাজিয়ে রাখাকেই ইনডেক্স বলে। ডেটাবেজ টেবিলের এক বা একাধিক ফিল্ডের উপর ইনডেক্স করে Alphabetically বা Numerically সাজানো যায়। ইনডেক্স ফাইল মূল ডেটাবেজ ফাইলের কোনরূপ পরিবর্তন না করে বিভিন্নভাবে সাজাতে পারে।

যেমনঃ নিচের Fact টেবিল থেকে যদি ২ রোল ধারীর GPA জানতে চাওয়া হয়। তাহলে কিন্তু একটু বেশি টাইম লাগবে। কারণ রোল ফিল্ডের মান গুলো সাজানো নেই। কিন্তু Fact টেবিলের ইনডেক্স থেকে ২ রোল ধারীর GPA খুব সহজেই পাওয়া যায়। কারণ রোল ফিল্ডের মান গুলো সাজানো আছে।

## Fact Table

Roll	Name	Section	GPA
3	Rafid	C	4.50
6	Akash	A	3.50
11	Monir	C	5.00
5	Ridoy	B	4.80
2	Babul	B	3.50
4	Nirob	A	4.00
7	Ashek	B	5.00
9	Obama	C	2.60
8	Fahim	A	3.10
1	Razu	A	5.00
10	Kamal	B	4.00

## Index Table

Roll	GPA
1	5.00
2	3.50
3	4.50
4	4.00
5	4.80
6	3.50
7	5.00
8	3.10
9	2.60
10	4.00
11	5.00

### ইনডেক্স করার সময় বিবেচ্য বিষয় সমূহঃ

- ১। সাধারণত কী ফিল্ডের উপর ইনডেক্স করতে হয় এবং ইনডেক্স এর একটি নাম দিতে হয়।
- ২। যে ফিল্ডের উপর ভিত্তি করে ইনডেক্স করা হয় সেই নামের অনুরূপ নাম নির্বাচন করতে হয়। এতে ইনডেক্স সমূহ মনে রাখতে সুবিধা হয়।
- ৩। এক বা একাধিক ফিল্ডের উপর ভিত্তি করে ইনডেক্স করা যায়। কোনো ডেটা টেবিলে এক বা একাধিক ইনডেক্স থাকতে পারে বা একই সময়ে খোলা থাকতে পারে। কিন্তু একই সময়ে কেবল একটি ইনডেক্স সক্রিয় থাকবে এবং রেকর্ডসমূহ প্রদর্শনের অর্ডার নিয়ন্ত্রণ করবে।

### ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

### ইনডেক্সিং এর সুবিধা সমূহঃ



- ১। ইনডেক্সিং এর বড় সুবিধা হলো ইনডেক্স তৈরি করার ফলে ফাইলে সহজে ডেটা খুঁজে বের করা যায়।
- ২। ইনডেক্স তৈরি করার ফলে ডেটাবেজ টেবিলে নতুন কোনো রেকর্ড ইনপুট করা হলেও ইনডেক্স ফাইলগুলো স্বয়ংক্রিয়ভাবে আপডেট হয়ে যায়।
- ৩। ডেটাসমূহের ইনডেক্স তৈরি করলে পারফরমেন্স ভাল পাওয়া যায়।
- ৪। ইনডেক্স ফাইল মূল ডেটাবেজ ফাইলের কোনোরূপ পরিবর্তন না করে বিভিন্নভাবে সাজাতে পারে।

### ইনডেক্সিং এর অসুবিধা সমূহঃ

- ১। যদি একাধিক ফিল্ডের উপর ইনডেক্সিং করা থাকে তাহলে কোনো ডেটা এডিট করলে ইনডেক্স ফাইল আপডেট করার জন্য দীর্ঘ সময় অপেক্ষা করতে হয়।
- ২। অনেক রেকর্ডের জন্য ইনডেক্স করা হলে অপেক্ষাকৃত বেশি মেমোরির প্রয়োজন হয়। তাছাড়া ইনডেক্স সংরক্ষণের জন্যও কিছু জায়গা লাগে।
- ৩। ডেটা এন্ট্রির ক্ষেত্রে ইনডেক্স ফাইলের রেফারেন্সসমূহ আপডেট হতে বেশ সময়ের প্রয়োজন হয়। সেজন্য ডেটা এন্ট্রি করতেও বেশি সময় লাগে।

### সর্টিং এবং ইনডেক্সিং এর মধ্যে পার্থক্য আলোচনাঃ

টেবিলের ডেটাগুলোর একটি ক্রম তৈরি করতে ইনডেক্সিং এবং সর্টিং উভয় পদ্ধতিই ব্যবহৃত হয়। ইনডেক্সিং একটি ইনডেক্স ফাইল তৈরি করে যা টেবিলের রেকর্ডগুলোর ফিজিক্যাল অবস্থানের পাশাপাশি সারিগুলোর লজিক্যাল ক্রম ধারণ করে, অপরদিকে একটি টেবিল সর্টিং করার সাথে সাথে টেবিলটির একটি সর্টেড অনুলিপি তৈরি হয়। সাধারণত, ইনডেক্স ফাইলটি একটি সর্টেড টেবিল সংরক্ষণের চেয়ে কম মেমোরির প্রয়োজন হয়। ইনডেক্সিং রেকর্ডগুলোর মূল ক্রম পরিবর্তন করে না, যেখানে সর্টিং রেকর্ডগুলোর মূল ক্রম পরিবর্তন করে। বাস্তবক্ষেত্রে সর্টিং এর চেয়ে ইনডেক্সিং বেশি ব্যবহৃত হয়।

#### ইনডেক্স তৈরি করার SQL কমান্ডের সিনট্যাক্স:

একটি টেবিলের উপর ইনডেক্স তৈরি করবে, যেখানে ডুপ্লিকেট ডেটা থাকতে পারেঃ

```
1 CREATE INDEX index_name
2 ON table_name (column1, column2, ...);
```

একটি টেবিলের উপর ইনডেক্স তৈরি করবে, যেখানে ডুপ্লিকেট ডেটা থাকবে নাঃ

```
1 CREATE UNIQUE INDEX index_name
2 ON table_name (column1, column2, ...);
```

#### ইনডেক্স ডিলিট করার SQL কমান্ডের সিনট্যাক্স:

```
1 DROP INDEX index_name ON table_name;
```

### ষষ্ঠ অধ্যায় পাঠ-৭: ডেটাবেজ মডেল এবং বিভিন্ন প্রকার কী ফিল্ড।

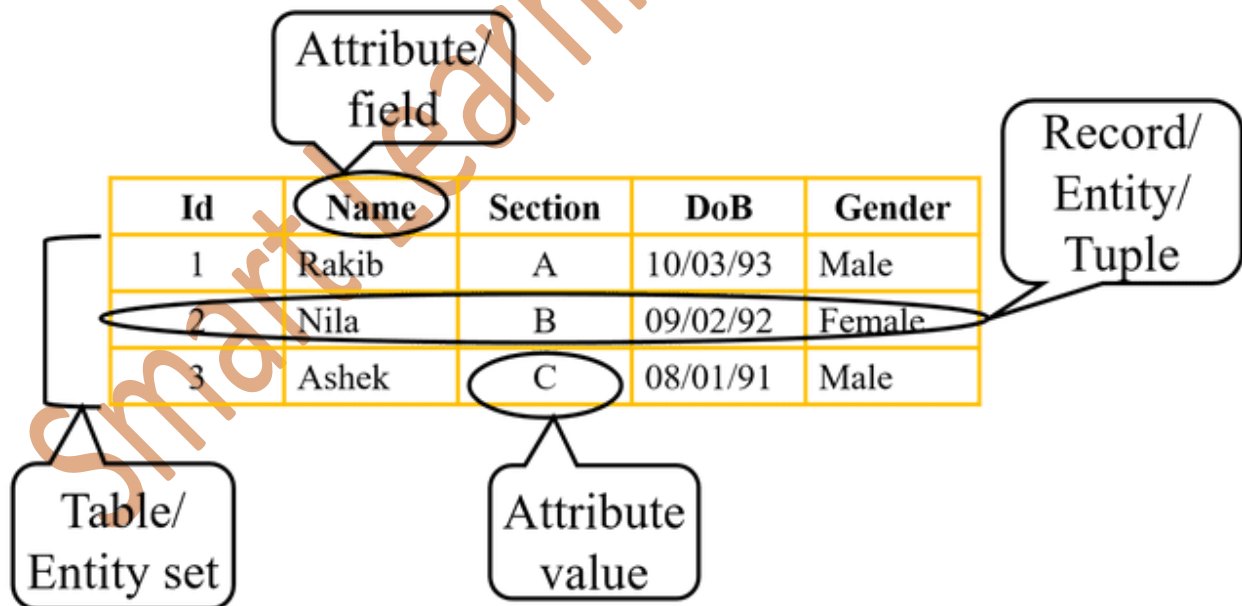
## ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**ডেটাবেজ মডেল:** ডেটাবেজ মডেল ডেটাবেজের লজিক্যাল ডিজাইন এবং স্ট্রাকচার নির্ধারণ করে এবং কোন ডেটাবেজ ম্যানেজমেন্ট সিস্টেমে কীভাবে তথ্য সংরক্ষণ, অ্যাক্সেস এবং আপডেট করা হবে তা নির্ধারণ করে। বিভিন্ন ধরনের ডেটাবেজ মডেলঃ

- ১। লিনিয়ার মডেল
- ২। ER(Entity Relationship) মডেল
- ৩। রিলেশনাল মডেল
- ৪। হায়ারার্কিক্যাল মডেল
- ৫। নেটওয়ার্ক মডেল
- ৬। অবজেক্ট ওরিয়েন্টেড ডেটাবেজ মডেল

**Entity Relationship মডেল:** ER মডেল হলো এমন একটি মডেল যা ডেটার মধ্যে সম্পর্ক উপস্থাপন এবং ডিজাইনের জন্য ব্যবহৃত হয়। এই ডেটাবেজ মডেলে, অবজেক্টকে এন্টিটি এবং এর বৈশিষ্ট্যকে অ্যাট্রিবিউটে ভাগ করে রিলেশনশিপ তৈরি করা হয়। ER মডেলটি বুঝার জন্য নিচের টার্মগুলো ভালোভাবে জানতে হবে।

- এনটিটি এবং এনটিটি সেট
- অ্যাট্রিবিউট এবং এর ধরণ
- কী বা কী ফিল্ড
- রিলেশনশিপ



**এনটিটি/রেকর্ড/ট্যাপল/সারি:** এনটিটি হলো সাধারণত বাস্তব-বিশ্বের এমন একটি বস্তু বা অবজেক্ট যার বিভিন্ন বৈশিষ্ট্য থাকে এবং DBMS এ একটি রিলেশনশিপ ধারণ করে। পরস্পর সম্পর্কযুক্ত একাধিক ফিল্ড নিয়ে গঠিত হয় এক একটি রেকর্ড। যেমন- একজন ছাত্র একটি এনটিটি।

**এনটিটি সেট/ টেবিলঃ** যদি একজন ছাত্র একটি এনটিটি হয়ে, তাহলে সকল ছাত্রের ডেটা সেটকে একত্রে এনটিটিসেট বলা হয়। এক বা একাধিক রেকর্ড নিয়ে টেবিল তৈরি হয়।

**অ্যাট্রিবিউট/ফিল্ড/কলাম :** কোনো একটি এনটিটি সেটের যে প্রোপার্টিগুলো ঐ এনটিটির বৈশিষ্ট্যগুলো প্রকাশ করে এবং যার ওপর ভিত্তি করে উপাত্ত গ্রহণ, প্রক্রিয়াকরণ ও সংরক্ষণ করা হয় তাকে অ্যাট্রিবিউট বলে। যেমন- একজন ছাত্র একটি এনটিটি যার অ্যাট্রিবিউট হলো Id, Name ইত্যাদি। অ্যাট্রিবিউটকে ভিজুয়াল ডেটাবেজ প্রোগ্রামে সাধারণত ডেটা ফিল্ড বলে। অ্যাট্রিবিউট বিভিন্ন ধরনের হয়ে থাকে। যেমন-

- **সাধারণ অ্যাট্রিবিউট:** এমন একটি অ্যাট্রিবিউট যার মান আর ক্ষুদ্র অংশে ভাঙা যায় না। উদাহরণস্বরূপ, ছাত্রের বয়স।
- **কম্পোজিট অ্যাট্রিবিউট:** একটি কম্পোজিট অ্যাট্রিবিউট একাধিক সাধারণ অ্যাট্রিবিউট এর সমন্বয়ে তৈরি। উদাহরণস্বরূপ, ছাত্রের ঠিকানা একটি কম্পোজিট অ্যাট্রিবিউট। যেখানে বাড়ি নম্বর, রাস্তার নাম, পিনকোড ইত্যাদি থাকবে।
- **ডিরাইভড অ্যাট্রিবিউট:** ডিরাইভড অ্যাট্রিবিউট সাধারণত ডেটাবেজ ম্যানেজমেন্ট সিস্টেমে উপস্থিত থাকে না, কিন্তু অন্যান্য অ্যাট্রিবিউট থেকে ডিরাইভড হয়। উদাহরণস্বরূপ, বয়স জন্ম তারিখ থেকে প্রাপ্ত হতে পারে।
- **সিঙ্গেল-ভ্যালুড অ্যাট্রিবিউট:** এই ধরনের অ্যাট্রিবিউটের সাধারণত সিঙ্গেল মান হয়ে থাকে।
- **মাল্টি-ভ্যালুড অ্যাট্রিবিউট:** এই ধরনের অ্যাট্রিবিউটের সাধারণত একাধিক মান থাকতে পারে। যেমন মোবাইল নম্বর।

**অ্যাট্রিবিউট ভেল্যু বা মান:** একটি এনটিটি সেটের প্রত্যেকটি অ্যাট্রিবিউটের একটি নির্দিষ্ট মান আছে। অ্যাট্রিবিউটের এ মানকে তার ভেল্যু বলে।

**কী বা কী ফিল্ডঃ** ডেটাবেজ টেবিলের রেকর্ড শনাক্তকরণ, অনুসন্ধান এবং ডেটাবেজের একাধিক টেবিলের মধ্যে সম্পর্ক স্থাপনের জন্য যে ফিল্ড ব্যবহার করা হয় তাকে বলা হয় কী ফিল্ড। কী ফিল্ডের ডেটাগুলো হবে অভিন্ন ও অদ্বিতীয়।

**রিলেশনশিপঃ** একটি ডেটাবেজের মধ্যে এক বা একাধিক টেবিল থাকে। এই টেবিলগুলোর মধ্যকার সম্পর্ককে ডেটাবেজ রিলেশনশিপ বলা হয়। ডেটাবেজে রিলেশনশিপ তৈরি করার জন্য যে কয়টি এনটিটি সেট বা টেবিল ব্যবহার করা হয় তার সংখ্যাকেই রিলেশনশিপের ডিগ্রি বলা হয়।

**রিলেশনশিপের ডিগ্রি সাধারণত তিন ধরনের হতে পারে। যথা:**

- ১। ডিগ্রি ১ বা ইউনারি রিলেশনশিপ
- ২। ডিগ্রি ২ বা বাইনারি রিলেশনশিপ
- ৩। ডিগ্রি ৩ বা টারনারি রিলেশনশিপ

**ডিগ্রি ১ বা ইউনারি রিলেশনশিপঃ** ইউনারি রিলেশনশিপে শুধু মাত্র একটি এনটিটি সেট/টেবিল অংশগ্রহণ করে। যেমন- মানুষ একটি এনটিটি। একজন মানুষ অন্য একজন মানুষকে বিয়ে করে। কাজেই মানুষ এনটিটি নিজের সাথে নিজের রিলেশনশিপ তৈরি করেছে।

**ডিগ্রি ২ বা বাইনারি রিলেশনশিপঃ** বাইনারি রিলেশনশিপে দু'টি এনটিটি সেট/টেবিল অংশগ্রহণ করে। যেমন- ছাত্র ও শিক্ষক দুটি পৃথক এনটিটি সেট বা টেবিল। শিক্ষক ছাত্রকে শিক্ষাদান করেন। কাজেই শিক্ষক এনটিটি ছাত্র এনটিটির সাথে বাইনারি রিলেশনশিপ তৈরি করেছে।

**ডিগ্রি ৩ বা টারনারি রিলেশনশিপঃ** টারনারি রিলেশনশিপে তিনটি এনটিটি সেট/টেবিল অংশগ্রহণ করে। যেমন- বিক্রেতা, পণ্য ও ওয়্যারহাউজ তিনটি পৃথক এনটিটি সেট বা টেবিল। বিক্রেতা ওয়্যারহাউজে পণ্য সরবরাহ করেন। কাজেই বিক্রেতা, পণ্য ও ওয়্যারহাউজ টারনারি রিলেশনশিপ তৈরি করেছে।

**ER Diagram:** একটি ডেটাবেজের বিভিন্ন টেবিলের মধ্যে সম্পর্ক বুঝানোর জন্য ER Diagram ব্যবহৃত হয়। ER Diagram এ ব্যবহৃত বিভিন্ন সিম্বল ও তাদের ব্যবহার-



এনটিটি সেটকে নির্দেশ করে।



এনটিটির বৈশিষ্ট্য বা অ্যাট্রিবিউট নির্দেশ করে।



মাল্টি-ভ্যালুড অ্যাট্রিবিউট নির্দেশ করে। যেমন-  
Mobile Number



ডিরাইভড অ্যাট্রিবিউট নির্দেশ করে। যেমন- Age



টেবিলগুলোর মধ্যে রিলেশন বুঝাতে ব্যবহৃত হয়।

নিচের টেবিলগুলো দেখা যাক-

### Student\_info

Roll	Name	DoB	Age
1	Rana	10/03/93	25
2	Raju	12/04/98	26

### Teacher\_info

Id	Name	Mobile
101	Rokib	017...12
102	Kashem	019...15

উপরের টেবিল দুটির মধ্যে রিলেশনশিপ নিচের ER diagram এর মাধ্যমে দেখানো হলঃ



ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**রিলেশনাল ডেটাবেজ মডেলঃ** রিলেশনাল ডেটাবেজ মডেল হলো আধুনিক ডেটাবেজ টেকনোলজির ভিত্তি। রিলেশনাল ডেটাবেজ মডেলে মূলত পুরো ডেটাবেজকে বিভিন্ন লজিক্যাল ইউনিটে বিভক্ত করা হয়। প্রতিটি লজিক্যাল ইউনিট হলো এক একটি টেবিল। প্রতিটি টেবিলে অনেকগুলো ফিল্ড থাকতে পারে। তবে প্রত্যেকটি টেবিলে একটি কী ফিল্ড থাকা বাঞ্ছনীয়। ডেটাবেজের টেবিলগুলো প্রাইমারি কী ও ফরেন কী এর মাধ্যমে পরস্পর সম্পর্কযুক্ত থাকে বিধায় এই মডেলকে রিলেশনাল ডেটাবেজ মডেল বলে।

নিচের চিত্রের মাধ্যমে বিভিন্ন মোবাইলের তথ্য নিয়ে তৈরি করা একটি রিলেশনাল ডেটাবেজ মডেল দেখানো হল।

Mld	BrandId	ModelId	ColorId	Price
1	1	2	2	25000
2	2	3	3	27000
3	3	1	1	15000

ColorId	Color
1	White
2	Black
3	Golden

ModelId	Model	BrandId
1	D620	3
2	J7	1
3	F9	2

BrandId	Brand
1	Samsung
2	Oppo
3	HTC

কী ফিল্ডঃ ডেটাবেজ টেবিলের রেকর্ড শনাক্তকরণ, অনুসন্ধান এবং ডেটাবেজের একাধিক টেবিলের মধ্যে সম্পর্ক স্থাপনের জন্য যে ফিল্ড ব্যবহার করা হয় তাকে বলা হয় কী ফিল্ড। কী ফিল্ডের ডেটাগুলো হবে অভিন্ন ও অদ্বিতীয়।  
নিচের টেবিলের “Roll” ফিল্ডটি কী ফিল্ড।

Roll	Name	Section	GPA	City
1	Romjan	2	5.00	Khulna
2	Manik	1	4.00	Dhaka
3	Tutul	1	3.50	Jamalpur
4	Manik	2	4.50	Nator
5	Rubel	1	4.00	Dhaka

কী ফিল্ড এর প্রকারভেদ-

- ১। ক্যান্ডিডেট কী
- ২। প্রাইমারি কী
- ৩। কম্পোজিট প্রাইমারি কী
- ৪। ফরেন কী

**ক্যান্ডিডেট কী:** ক্যান্ডিডেট কী হলো টেবিলের একটি কলাম বা কলামের সেট যা কোনও ডেটাবেস রেকর্ডকে অদ্বিতীয়ভাবে সনাক্ত করতে পারে। প্রতিটি টেবিলে এক বা একাধিক ক্যান্ডিডেট কী থাকতে পারে, তবে একটি ক্যান্ডিডেট কী কে প্রাইমারি কী বলা হয়। নিচের টেবিলে “Roll” এবং “NID” কে ক্যান্ডিডেট কী বলা হয়।

Roll	Name	NID	City
1001	Kamrul	1993..12	Khulna
1002	Rokib	1993..13	Khulna
1003	Kamrul	1993..14	Dhaka

**প্রাইমারি কী (Primary Key):** কোনো ডেটাবেজ টেবিলের যে ফিল্ডের প্রতিটি ডেটা অদ্বিতীয় (Unique) এবং যার সাহায্যে টেবিলের সবগুলো রেকর্ডকে অদ্বিতীয়ভাবে সনাক্ত করা যায় তাকে প্রাইমারি কী বলা হয়। প্রাইমারি কী এর সাহায্যে এক বা একাধিক টেবিলের মধ্যে সম্পর্ক স্থাপন করে সম্পর্কযুক্ত ডেটাবেজ তৈরি করা যায়। একটি টেবিল তৈরি করার সময়ই প্রাইমারি কী নির্ধারণ করা হয়। নিচের ডেটাবেজ টেবিলের Roll ফিল্ডের প্রতিটি মান অদ্বিতীয় তাই এই ফিল্ডকে প্রাইমারি কী বলা হয়।

Roll	Name	City
1001	Kamrul	Khulna
1002	Rokib	Khulna
1003	Kamrul	Dhaka

প্রাইমারি কী – এর বৈশিষ্ট্যসমূহ হলো-



- ১। একটি টেবিলে একটির বেশি প্রাইমারি কী থাকতে পারবে না।
- ২। প্রাইমারি কী-তে একই value একাধিকবার এবং Null Value থাকতে পারে না।
- ৩। একাদিক টেবিলের মধ্যে রিলেশন তৈরি করার পর কোন টেবিলের প্রাইমারি কী পরিবর্তন করা যায় না।

**কম্পোজিট প্রাইমারি কী:** একাধিক ফিল্ডের সমন্বয়ে যে প্রাইমারি কী গঠন করা হয় তাকে কম্পোজিট প্রাইমারি কী বলা হয়। রিলেশনাল ডেটাবেজ মডেলে অংশগ্রহণকারী টেবিলের ক্ষেত্রে যদি এমন হয় যে, একটি টেবিলের কোন একটি ফিল্ডের প্রতিটি ডেটা অদ্বিতীয় নেই। তখন একাদিক ফিল্ডের সমন্বয়ে প্রাইমারি কী গঠন করা হয়। নিচের টেবিলে Roll এবং Section একত্রে কম্পোজিট প্রাইমারি গঠন করতে পারে।

Roll	Name	Section	City
1	Kamrul	A	Khulna
2	Rokib	A	Khulna
1	Nirob	B	Jamalpur
2	Ridoy	B	Sherpur

ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**ফরেন কী:** রিলেশনাল ডেটাবেজ মডেলে কোনো একটি টেবিলের প্রাইমারি কী যদি অন্য টেবিলে ব্যবহৃত হয় তখন ঐ কী কে প্রথম টেবিলের সাপেক্ষে দ্বিতীয় টেবিলের ফরেন কী বলে। ফরেন কী এর সাহায্যে একটি টেবিলের সাথে অন্য টেবিলের সম্পর্ক স্থাপন করা যায়। যেমন- নিচের চিত্রে Subject\_info টেবিলের S\_id প্রাইমারি কী Teacher\_info টেবিলে ব্যবহৃত হয়েছে। তাই Teacher\_info টেবিলের ক্ষেত্রে S\_id ফিল্ডটি ফরেন কী।

## Teacher\_info

T_id	T_Name	S_Code
T001	Kamrul	101
T002	Rokib	275
T003	Kamrul	174
T004	Nirob	275
T005	Ridoy	178

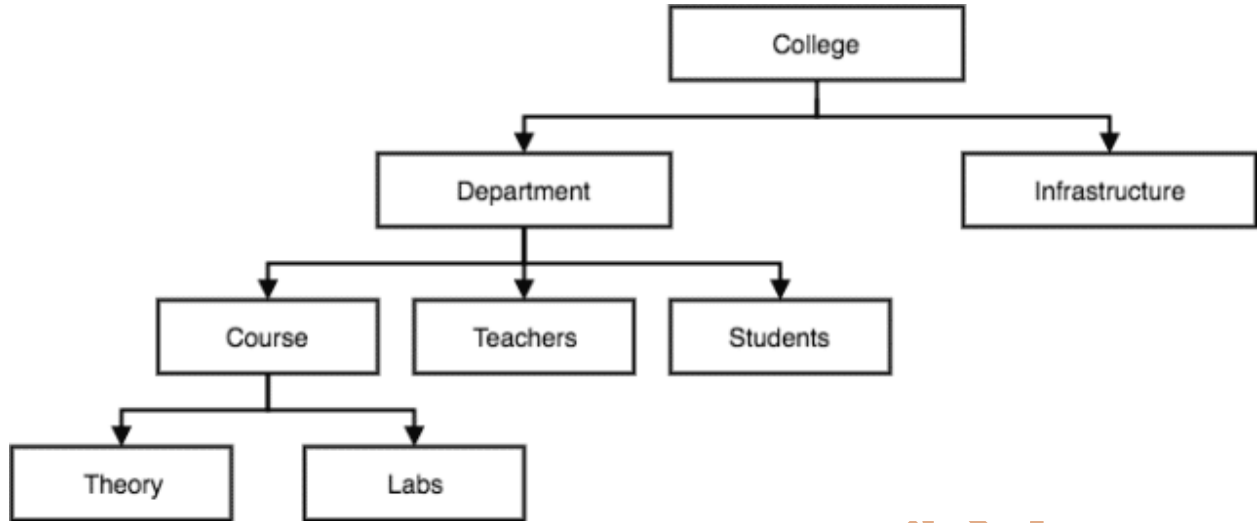
## Subject\_info

S_Code	S_Name
101	Bangla
107	English
275	ICT
174	Physics
176	Chemistry
265	Math
178	Biology

ফরেন কী -এর বৈশিষ্ট্যসমূহ হলো:

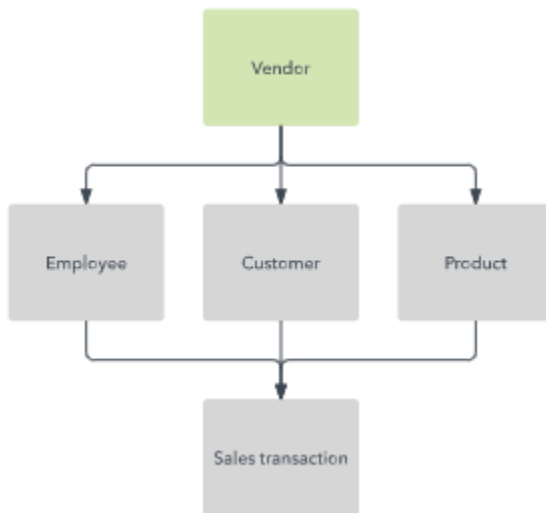
- ১। একটি টেবিলের ফরেন কী (Foreign Key) কে অবশ্যই রেফারেন্স টেবিলের (Reference table) প্রাইমারি কী অথবা ইউনিক কী হতে হবে।
- ২। ফরেন কী ফিল্ডের ভেল্যু অবশ্যই রেফারেন্স টেবিলের প্রাইমারি কী এর ভেল্যু হতে হবে।
- ৩। ডুপ্লিকেট (Duplicate) অথবা নাল (Null) ভেল্যু ইনসার্ট (Insert) করা যাবে।

**হায়ারার্কিক্যাল মডেলঃ** হায়ারার্কিক্যাল মডেলটি ট্রি বা গাছের মতো কাঠামোতে তথ্য সংগঠিত করে, যেখানে প্রতিটি রেকর্ডের একটি রুট রেকর্ড থাকে। চাইল্ড রেকর্ডগুলো একটি নির্দিষ্ট অর্ডারে সাজানো থাকে। ডেটাবেজে ডেটা স্টোর করার জন্য সেই অর্ডারটি ফিজিক্যাল অর্ডার হিসেবে ব্যবহৃত হয়। বাস্তব-বিশ্বের অনেক সম্পর্ককে বর্ণনা করার জন্য এই মডেলটি ভাল।



এই মডেলটি প্রাথমিকভাবে আইবিএম (IBM) এর তথ্য ব্যবস্থাপনা সিস্টেমে 60 এবং 70 এর দশকে ব্যবহৃত হতো, তবে কিছু কার্য অক্ষমতার কারণে আজকাল খুব কমই এটি দেখা যায়।

**নেটওয়ার্ক মডেলঃ** নেটওয়ার্ক মডেলটি হায়ারার্কিক্যাল মডেলের উপর ভিত্তি করে তৈরি। এই মডেলটি সংযুক্ত রেকর্ডগুলোর মধ্যে many-to-many রিলেশনশিপের অনুমতি দেয় এবং প্রতিটি চাইল্ড রেকর্ডের একাধিক রুট রেকর্ড থাকতে পারে। গাণিতিক সেট তত্ত্বের উপর ভিত্তি করে, এই মডেলটি সম্পর্কযুক্ত রেকর্ডগুলোর সেটের সমন্বয়ে তৈরি। প্রতিটি সেট একটি পেরেন্ট বা রুট রেকর্ড এবং একাধিক চাইল্ড রেকর্ডের সমন্বয়ে তৈরি। একটি রেকর্ড একাধিক সেটের সদস্য বা চাইল্ড হতে পারে, এই মডেল জটিল রিলেশনশিপ প্রকাশ করতে পারে।



এই মডেলটি ডেটা সিস্টেম ভাষা (CODASYL) এর সম্মেলন দ্বারা আনুষ্ঠানিকভাবে সংজ্ঞায়িত হওয়ার পরে 70 এর দশকের মধ্যে সবচেয়ে জনপ্রিয় ছিল।

## ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

### ষষ্ঠ অধ্যায় পাঠ-৮ : ডেটাবেজ রিলেশন এবং এর বিভিন্ন প্রকারভেদ।

**ডেটাবেজ রিলেশনঃ** ডেটাবেজের একটি টেবিলের রেকর্ডের সাথে অপর একটি টেবিলের রেকর্ডের সম্পর্কে ডেটাবেজ রিলেশন বলে। অর্থাৎ ডেটাবেজ রিলেশন হলো বিভিন্ন ডেটা টেবিলের মধ্যকার লজিক্যাল সম্পর্ক।

#### ডেটাবেজ রিলেশনের শর্তঃ

- ১। রিলেশনাল ডেটা টেবিলগুলোর মধ্যে কমপক্ষে একটি কমন ফিল্ড থাকবে। কমন ফিল্ডের ডেটা টাইপ, ফিল্ড সাইজ এবং ফরমেট ইত্যাদি একই হতে হবে।
- ২। রিলেশনাল টেবিলগুলোর মধ্যে অন্তত একটি টেবিলে অবশ্যই প্রাইমারি কী ফিল্ড থাকতে হবে।

#### ডেটাবেজ রিলেশনের প্রকারভেদঃ

১. One to One রিলেশন
২. One to Many রিলেশন
৩. Many to One রিলেশন
৪. Many to Many রিলেশন

**One to One রিলেশনঃ** যদি ডেটাবেজের একটি টেবিলের একটি রেকর্ড অপর একটি ডেটা টেবিলের কেবল মাত্র একটি রেকর্ডের সাথে সম্পর্কিত থাকে তবে তাদের মধ্যকার রিলেশনকে বলা হয় One to One রিলেশন।

ধরা যাক একটি কলেজে যথেষ্ট পরিমাণ শিক্ষক আছে। তাই কার্য পরিচালনা পর্ষদ নিয়ম করল যে; একজন শিক্ষক শুধুমাত্র একটি সাবজেক্টের ক্লাস নিতে পারবে এবং যেকোনো একটি সাবজেক্ট একজন শিক্ষক শুরু করলে তাকেই শেষ করতে হবে অর্থাৎ অন্য কোনো শিক্ষক এই সাবজেক্ট পড়াতে পারবে না।

#### Teacher\_info

T_Id	Name
T001	Rokib
T002	Kashem
T003	Ridoy
T004	Prantor

#### Subject\_info

T_Id	Subject_code	Subject
T001	101	Bangla
T002	108	English
T003	275	ICT
T004	174	Physics

উপরের টেবিল থেকে দেখা যাচ্ছে, একজন শিক্ষক কেবলমাত্র একটি সাবজেক্ট এর ক্লাস নিচ্ছে। অর্থাৎ Teacher\_info টেবিলের একটি রেকর্ডের সাথে Subject\_info টেবিলের কেবলমাত্র একটি রেকর্ড সম্পর্কিত। তাই তাদের মধ্যে One to One রিলেশন বিদ্যমান।

**One to many রিলেশনঃ** যদি ডেটাবেজের একটি টেবিলের একটি রেকর্ড অপর একটি ডেটা টেবিলের একাধিক রেকর্ডের সাথে সম্পর্কিত থাকে তবে তাদের মধ্যকার রিলেশনকে বলা হয় One to many রিলেশন।

পূর্বের কলেজটির কথাই চিন্তা করা যাক। ধরা যাক সেই কলেজ থেকে কিছু শিক্ষক চলে গেলেন। তাই এখন নতুন নিয়ম করা হলো, একজন শিক্ষক একাধিক সাবজেক্টের ক্লাস নিতে পারবে, কিন্তু একটি সাবজেক্ট একাধিক শিক্ষক নিতে পারবে না।

### Teacher\_info

T_Id	Name
T001	Rokib
T002	Kashem
T003	Ridoy
T004	Prantor

### Subject\_info

T_Id	Subject_code	Subject
T001	101	Bangla
T001	108	English
T003	275	ICT
T004	174	Physics

উপরের টেবিল থেকে দেখা যাচ্ছে, একজন শিক্ষক একাধিক সাবজেক্ট এর ক্লাস নিচ্ছে। অর্থাৎ Teacher\_info টেবিলের একটি রেকর্ডের সাথে Subject\_info টেবিলের একাধিক রেকর্ড সম্পর্কিত। তাই তাদের মধ্যে One to many রিলেশন বিদ্যমান।

## ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**Many to One রিলেশনঃ** যদি ডেটাবেজের একটি টেবিলের একাধিক রেকর্ড অপর একটি ডেটা টেবিলের একটি মাত্র রেকর্ডের সাথে সম্পর্কিত থাকে তবে তাদের মধ্যকার রিলেশনকে বলা হয় Many to One রিলেশন।

আরও এক বছর পর ঐ কলেজে প্রচুর সংখ্যক নতুন শিক্ষক এর নিয়োগ দেওয়া হলো। তাই এবার [নিয়ম](#) করা হলো একজন শিক্ষক একটির বেশি সাবজেক্টের ক্লাস নিতে পারবে না। কিন্তু একটি সাবজেক্ট অনেকজন শিক্ষক ভাগ করে নিতে পারবে।

## Teacher\_info

T_Id	Name	Subject_code
T001	Rokib	101
T002	Kashem	101
T003	Ridoy	275
T004	Prantor	174

## Subject\_info

Subject_code	Subject
101	Bangla
108	English
275	ICT
174	Physics

উপরের টেবিল থেকে দেখা যাচ্ছে, একাধিক শিক্ষক একটি সাবজেক্ট এর ক্লাস নিচ্ছে। অর্থাৎ Teacher\_info টেবিলের একাধিক রেকর্ডের সাথে Subject\_info টেবিলের একটি রেকর্ড সম্পর্কিত। তাই তাদের মধ্যে Many to One রিলেশন বিদ্যমান।

**Many to Many রিলেশনঃ** যদি ডেটাবেজের একটি টেবিলের একাধিক রেকর্ড অপর একটি ডেটা টেবিলের একাধিক রেকর্ডের সাথে সম্পর্কিত থাকে তবে তাদের মধ্যকার রিলেশনকে বলা হয় Many to Many রিলেশন।

আরও কিছুদিন পর বোর্ড থেকে সিলেবাস পরিবর্তন করার ফলে সাজেটের সংখ্যা বেড়ে গেল। তাই শিক্ষকের সংখ্যা অনুপাতে কম হওয়ায় নতুন নিয়ম করা হলো; একজন শিক্ষককে একাধিক সাবজেক্টের ক্লাস নিতে হতে পারে আবার একটি সাবজেক্ট একাধিক শিক্ষককে ভাগাভাগি করে নিতে হতে পারে।

T_Id	Name
T001	Rokib
T002	Kashem
T003	Ridoy
T004	Prantor

Teacher\_info

Subject_code	Subject
101	Bangla
108	English
275	ICT
174	Physics

Subject\_info

## Joining\_info

T_Id	Subject_code	Day
T001	101	Sat
T002	101	Sun
T003	275	Mon
T003	174	Tue

many to many রিলেশনের বৈশিষ্ট্যঃ

- ১। এই রিলেশন প্রতিষ্ঠা করার জন্য অতিরিক্ত একটি তৃতীয় টেবিলের প্রয়োজন হয় যাকে জাংশন টেবিল বলে।
- ২। জাংশন টেবিলে উভয় টেবিলের প্রাইমারী কী দুটি ফরেন কী তৈরি করতে হয়। ফলে একটি টেবিলের প্রাইমারী কী জাংশন টেবিলের একটি ফরেন কী এর সাথে One to many রিলেশন তৈরি করে।
- ৩। অনুরূপ অপর টেবিলের প্রাইমারী কী জাংশন টেবিলের অপর ফরেন কী এর সাথে One to many রিলেশন তৈরি করে।
- ৫। ফলে জাংশন টেবিলের মাধ্যমে উক্ত টেবিলদ্বয়ে many to many রিলেশন তৈরি হয়।

### ষষ্ঠ অধ্যায় পাঠ-৯ কর্পোরেট ডেটাবেজ এবং সরকারি প্রতিষ্ঠানে ডেটাবেজ।

**কর্পোরেট ডেটাবেজ:** কর্পোরেট ডেটাবেজ হলো প্রতিষ্ঠানিক পর্যায়ে সেই ডেটাবেজ যা কোনো প্রতিষ্ঠান কর্তৃক বাণিজ্যিক ভিত্তিতে ব্যবহৃত হয়। প্রতিটি ব্যবসা প্রতিষ্ঠানের বিভিন্ন ধরনের বিভাগ বা অনুবিভাগ থাকে যেমন- উৎপাদন, বিক্রয়, বিতরণ, বিপণন, গ্রাহক সেবা, মানব সম্পদ উন্নয়ন, আইটি, অডিট, গ্রাহক সেবা ইত্যাদি। উক্ত প্রতিষ্ঠানের আওতায় থাকা সকল ডেটার এক বিশাল সংগ্রহ থাকে। বিভিন্ন প্রতিষ্ঠানকে তাদের কর্মী, ক্রেতা, সরবরাহকারি, অংশীদার এবং সরকারকে নির্ভুল ও সময়মতো তথ্য প্রদান করতে হয়। কর্পোরেট ডেটাবেজ এ কাজটিকে সহজ করে তোলে। কর্পোরেট ডেটাবেজে প্রতিষ্ঠানের তথ্যসমূহ সংরক্ষিত থাকে এবং পরবর্তীতে বিভিন্ন কাজের জন্য এ ডেটাবেজকে ব্যবহার করা হয়।

**কর্পোরেট পর্যায়ে ব্যবহৃত জনপ্রিয় কিছু ডেটাবেজ সফটওয়্যার হলো-**

- (i) Oracle
- (ii) DB2
- (iii) SQL Server
- (iv) Sybase
- (v) Teradata
- (vi) ADABAS
- (vii) MySQL
- (viii) FileMaker
- (ix) Access
- (x) Inform-ix

**ডাটাবেজ ম্যানেজমেন্ট সিস্টেম**

**সরকারি প্রতিষ্ঠানে ডেটাবেজ:** সরকারি প্রতিষ্ঠানের যাবতীয় উপাত্ত ও তথ্য সংরক্ষণের উপযুক্ত উপায় হলো সংশ্লিষ্ট বিষয়ে ডেটাবেজ তৈরি করা। শক্তিশালী সরকার পরিচালনা ব্যবস্থায় ডেটাবেজ হলো গুরুত্বপূর্ণ একটি উপাদান। প্রতিটি সরকারের অধীনে থাকে অসংখ্য মন্ত্রণালয়। এসব মন্ত্রণালয় স্বতন্ত্রভাবে কিংবা অন্যান্য মন্ত্রণালয়ের সাথে সমন্বিত উপায়ে কাজ করে থাকে। এসব কাজে তথ্যের ব্যবহার অপরিহার্য। এ তথ্য ব্যবস্থাপনাকে আরও সুন্দর ও কার্যোপযোগী করে তুলতে পারে ডেটাবেজ।

**সরকারি প্রতিষ্ঠানে ডেটাবেজ এর ব্যবহারঃ**



১। অপরাধমূলক কর্মকাণ্ডে জড়িত ব্যক্তিবর্গের ছবিসহ ব্যক্তিগত নানা তথ্য, আঙুলের ছাপ ইত্যাদি সংরক্ষণ করে রাখতে পারে দেশের আইনশৃঙ্খলা রক্ষাকারি বাহিনী। এর ফলে যেকোনো অপরাধ সংঘটিত হলে ঘটনাস্থলের বিভিন্ন আলামত, আঙুলের ছাপ, রক্ত, ব্যবহৃত হাতিয়ার ইত্যাদি পরীক্ষা করে সহজে অপরাধী সনাক্ত করা যায়।

২। সরকারি নানা গবেষণামূলক কার্যক্রম পরিচালনাকারি প্রতিষ্ঠানে সংশ্লিষ্ট তথ্য সংরক্ষণ করা, পরিসংখ্যান ব্যুরো, নির্বাচন কমিশন, ব্যানবেইজ প্রভৃতি প্রতিষ্ঠানে তথ্য প্রক্রিয়াকরণ ও সংরক্ষণ করা, আদমশুমারি, কৃষিশুমারি, অর্থনৈতিক সংক্রান্ত তথ্য সংরক্ষণ, বিবাহ, তালাক প্রভৃতি রেজিস্ট্রেশন ও রেকর্ড সংরক্ষণ।

৩। জন্মহার ও মৃত্যুহার নির্ণয়, জন্ম-মৃত্যু রেকর্ড সংরক্ষণ, দুর্ঘটনার রেকর্ড সংরক্ষণ, আইন, আদালত, মামলা, অধ্যাদেশ ইত্যাদি সংক্রান্ত রেকর্ড সংরক্ষণ।

৪। তথ্য ও ছবি সংগ্রহ করে নাগরিকদের ছবিসহ ভোটার আইডি সংরক্ষণ, বাড়ির হোল্ডিং নম্বর, ভূমি ট্যাক্স, আয়কর, ফোন নম্বর সংরক্ষণ।

৫। পাবলিক বিশ্ববিদ্যালয়ের পরীক্ষাগুলোর ছাত্রদের ফলাফল ও তথ্য সংরক্ষণ করা হয় এর ফলে আমরা যেকোনো সময় বিগত পাবলিক পরীক্ষার ফলাফল দেখতে পারি বা সংগ্রহ করতে পারি।

৬। শিক্ষার হার, পাসের হার, শিক্ষা প্রতিষ্ঠানের সংখ্যা ইত্যাদি সংরক্ষণ।

৭। ভূমি জরিপ, নানা ধরনের রেকর্ডের তথ্য সংরক্ষণ, সরকারি বিভিন্ন নথিপত্র ও জরিপ সংক্রান্ত রেকর্ড সংরক্ষণ, শহর বা গ্রামাঞ্চলভিত্তিক বিভিন্ন রেকর্ড সংরক্ষণ, জেলা, থানা বা এলাকাভিত্তিক বিভিন্ন তথ্য সংরক্ষণ, সামরিক বাহিনীর ভূমি, সরকারি বেসরকারি আয়-ব্যয়, রাজস্ব বা উন্নয়ন বরাদ্দ, বাজেট প্রভৃতি কাজে।

৮। তাছাড়া বর্তমানে অনলাইনে বিদেশ গমনেচ্ছুদের ডেটাবেজ সংরক্ষিত আছে। এর ফলে অতি সহজে নির্দিষ্ট পেশার বিদেশ গমনেচ্ছু শ্রমিক নির্বাচন করা যায়। মোট উৎপাদন, বার্ষিক জিডিপি, গড় আয়, বিদেশি ঋণের পরিমাণ প্রভৃতি নিরূপণ ও সংরক্ষণ, স্টক মার্কেটে শেয়ার দর, কোম্পানি প্রোফাইল, কোম্পানি প্রসপেক্টাস, সূচক প্রভৃতি নির্ণয় ও সংরক্ষণ আয়কর, কাস্টম, আমদানিরপ্তানি, রেমিট্যান্স প্রভৃতি সংরক্ষণ। অর্থাৎ সার্বিকভাবে সরকারি পর্যায়ে ডেটাবেজ ব্যবহারের কারণে রাষ্ট্রের কার্যক্রমে আরও গতিশীলতা আনা সম্ভব।

ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**ষষ্ঠ অধ্যায় পাঠ-১০: ডেটাবেজ সিকিউরিটি।**

**ডেটাবেজ সিকিউরিটিঃ** একটি ডেটাবেজে অনির্দিষ্ট ব্যবহারকারী থেকে ডেটা সুরক্ষিত রাখাকে বলা হয় ডেটাবেজ সিকিউরিটি।

**ডেটাবেজ সিকিউরিটি নিচের বিষয়গুলোকে নিয়ন্ত্রণ করে:**

১। ব্যবহারকারীর ডেটা ব্যবহার করার অধিকার সংরক্ষণ করা।

২। সিস্টেম রিসোর্স ব্যবহার নিয়ন্ত্রণ করা।

৩। ডিস্ক ব্যবহার নিয়ন্ত্রণ করা।

- ৪। ব্যবহারকারীর অ্যাকশন নিয়ন্ত্রণ করা।
- ৫। ব্যবহারকারীর ডেটা ব্যবহারের সীমা নির্ধারণ করা।

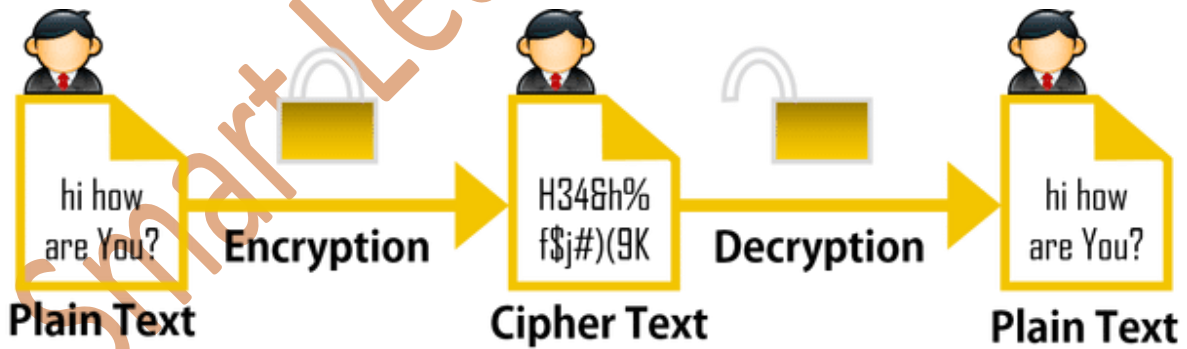
ডেটাবেজের সিকিউরিটি প্রধানত ২ ভাগে ভাগ করা যায়:

- ক। সিস্টেম সিকিউরিটি
- খ। ডেটা সিকিউরিটি

**সিস্টেম সিকিউরিটি:** ডেটাবেজের সিস্টেম লেভেলে অনির্দিষ্ট ব্যক্তির প্রবেশ রক্ষা করার জন্য গৃহীত ব্যবস্থাকে বলা হয় সিস্টেম সিকিউরিটি। সার্ভার কম্পিউটার অচল হয়ে গেলে ডেটাবেজের ডেটা হারিয়ে যায়। কিন্তু সিস্টেম সিকিউরিটি থাকলে ডেটা পুনরুদ্ধার করা যায়।

**ডেটা সিকিউরিটি:** অনির্দিষ্ট ব্যক্তির হাত থেকে ডেটার গোপনীয়তা রক্ষা করাকে বলা হয় ডেটা সিকিউরিটি। ডেটা সিকিউরিটির জন্য প্রাপককে ডেটা এনক্রিপ্ট করে পাঠানো হয়। প্রাপকের কাছে ডেটা পৌঁছানোর পর প্রাপক ডেটাকে ডিক্রিপ্ট করে তারপর ব্যবহার করে। ডেটাকে এনক্রিপশন ও ডিক্রিপশন করার বিষয়কে ক্রিপ্টোগ্রাফী বলে।

**ডেটা এনক্রিপশন এবং ডিক্রিপশন:** যে প্রক্রিয়ায় প্লেইনটেক্সটকে পরিবর্তন করে ছাইফারটেক্সট তৈরি করা হয় তাকে এনক্রিপশন বলে। যে প্রক্রিয়ায় ছাইফারটেক্সটকে পরিবর্তন করে পুনরায় প্লেইনটেক্সট তৈরি করা হয় তাকে ডিক্রিপশন বলে। উৎস ডেটাকে এনক্রিপ্ট করে পাঠালে প্রাপককে ঐ ডেটা ব্যবহারের পূর্বে ডিক্রিপ্ট করতে হয়। প্রেরক এবং প্রাপককে যথাক্রমে এনক্রিপ্ট এবং ডিক্রিপ্ট করার পদ্ধতি/ অ্যালগরিদম জানতে হয়।



ডেটা এনক্রিপশনের প্রধান চারটি অংশ:

- ১। প্লেইনটেক্সট (Plain Text)
- ২। সাইফারটেক্সট (Cipher-text)
- ৩। এনক্রিপশন অ্যালগরিদম (Encryption Algorithm)
- ৪। সিকিউরিটি কী বা কোড (Security key or Code)

**প্লেইনটেক্সট (Plain Text):** এনক্রিপ্ট করার পূর্বের ডেটা যা পাঠ করা যায় তাকে প্লেইনটেক্সট বলে।

**সাইফারটেক্সট (Cipher-text):** এনক্রিপ্ট করার পরের ডেটা যা পাঠ করা যায় না তাকে সাইফারটেক্সট বলে।

**এনক্রিপশন অ্যালগরিদম (Encryption Algorithm):** যে গাণিতিক ফর্মুলার মাধ্যমে প্লেইনটেক্সট থেকে সাইফারটেক্সট আবার সাইফারটেক্সট থেকে প্লেইনটেক্সট এ রূপান্তর করা হয় তাকে এনক্রিপশন অ্যালগরিদম বলে।

**সিকিউরিটি কী বা কোড (Security key or Code):** যে গোপন সংকেত বা কোডের মাধ্যমে ডেটা এনক্রিপ্ট ও ডিক্রিপ্ট করা হয় তাকে সিকিউরিটি কী বা কোড।

ডাটাবেজ ম্যানেজমেন্ট সিস্টেম

**ডেটা এনক্রিপ্ট করার বিভিন্ন পদ্ধতিঃ**

- ১। সিজার কোড (Caesar Code)
- ২। ডেটা এনক্রিপশন স্ট্যান্ডার্ড (Data Encryption Standard-DES)
- ৩। ইন্টারন্যাশনাল ডেটা এনক্রিপশন অ্যালগরিদম (International Data Encryption Algorithm-IDEA)

**এনক্রিপশন পদ্ধতি-১:**

এ পদ্ধতিতে ইংরেজি প্রত্যেক বর্ণের জন্য নির্দিষ্ট ক্রম অনুযায়ী নির্দিষ্ট বর্ণ ব্যবহার করা হয়। যেমন- ইংরেজি প্রত্যেক বর্ণকে তার পরবর্তী বর্ণ দ্বারা প্রতিস্থাপন করলে- ICT শব্দটির এনক্রিপশন হলো JDU। এখানে মূল শব্দের প্রত্যেক বর্ণের পরবর্তী বর্ণ ব্যবহার করে এনক্রিপ্ট করা হয়েছে।

**এনক্রিপশন পদ্ধতি-২:**

- ১। এ পদ্ধতিতে মূল ডেটার প্রত্যেক বর্ণকে ইংরেজি বর্ণমালার ক্রম অনুসারে অবস্থান নির্ণয় করা হয়। যেমন- A এর অবস্থানগত মান 1 এবং C এর অবস্থানগত মান 3।
- ২। অবস্থানগত সংখ্যাকে ৮ দ্বারা গুণ করা হয়।
- ৩। গুণফলের মানকে অবস্থান ধরে বর্ণমালার ক্রমানুসারে যে বর্ণটি পাওয়া যায় তা এনক্রিপ্টেড বর্ণ হিসাবে ধরা হয়।
- ৪। গুণফল ২৬ অপেক্ষা বড় হলে গুণফলকে ২৬ দ্বারা ভাগ করে ভাগশেষ নির্ণয় করা হয়। এক্ষেত্রে ভাগশেষের মানকে অবস্থান ধরে বর্ণমালার ক্রমানুসারে যে বর্ণটি পাওয়া যায় তা এনক্রিপ্টেড বর্ণ হিসাবে ধরা হয়।

**এ পদ্ধতিতে CAESAR শব্দটি এনক্রিপ্ট করি-**

$$\begin{array}{rclcl}
 C & = & 3 \times 8 & = 24 & \rightarrow X \\
 A & = & 1 \times 8 & = 8 & \rightarrow H \\
 E & = & 5 \times 8 & = 40 & \rightarrow 40 \div 26 \rightarrow \text{ভাগশেষ } 14 \rightarrow N \\
 S & = & 19 \times 8 & = 152 & \rightarrow 152 \div 26 \rightarrow \text{ভাগশেষ } 22 \rightarrow V \\
 A & = & 1 \times 8 & = 8 & \rightarrow H \\
 R & = & 18 \times 8 & = 144 & \rightarrow 144 \div 26 \rightarrow \text{ভাগশেষ } 14 \rightarrow N
 \end{array}$$

সুতরাং **CAESAR** শব্দটি এনক্রিপ্ট হয়ে **XHNVHN** হয়ে গেল, যা **Cipher-text** হিসেবে পরিচিত।