

Class #11: Introduction to SQL - Part 1

1. Overview of Relational Database Management Systems (RDBMS).
2. Database Normalization concepts.
3. Introduction to NoSQL databases.
4. Selecting data with SQL statements.
5. Filtering data using the WHERE clause and combining conditions.
6. Sorting data with keywords such as DISTINCT, TOP, LIKE, etc.
7. Modifying data using SQL syntax. Assignment #11:

1. Overview of Relational Database Management Systems (RDBMS).

RDBMS এর সংজ্ঞা: রিলেশনাল ডেটাবেস ম্যানেজমেন্ট সিস্টেম (RDBMS) হল একটি ডেটাবেস ম্যানেজমেন্ট সিস্টেম যা ডেটাকে টেবিল আকারে সংরক্ষণ করে। এই টেবিলগুলোতে ডেটা সারি (rows) এবং কলাম (columns) আকারে সংগঠিত থাকে। RDBMS সাধারণত SQL (Structured Query Language) ব্যবহার করে ডেটা সংরক্ষণ, পরিচালনা, এবং অনুসন্ধান (query) করে। RDBMS এর মূল বৈশিষ্ট্যগুলো হল:

1. Table: ডেটা টেবিল আকারে সংরক্ষিত থাকে, যেখানে প্রতিটি টেবিলের মধ্যে সারি এবং কলাম থাকে।
2. Relationship: টেবিলগুলোর মধ্যে সম্পর্ক থাকে, যা সাধারণত Primary Key (primary key) এবং Foreign Key (foreign key) ব্যবহার করে তৈরি হয়। Primary Key একটি টেবিলের প্রতিটি রেকর্ডের ইউনিক চিহ্ন, এবং Foreign Key অন্য টেবিলের রেকর্ডের সাথে সম্পর্ক স্থাপন করে।
3. Normalization: RDBMS ডেটার অতিরিক্ত পুনরাবৃত্তি কমাতে এবং ডেটার নির্ভরতা বজায় রাখতে Normalization প্রক্রিয়া অনুসরণ করে।
4. ACID গুণাবলী: RDBMS ডেটাবেস ট্রানজেকশনগুলোকে নিরাপদ এবং নির্ভরযোগ্য রাখতে ACID (Atomicity, Consistency, Isolation, Durability) principle অনুসরণ করে।
5. SQL (Structured Query Language): SQL ব্যবহার করে RDBMS ডেটা সংরক্ষণ, আপডেট, মুছা এবং অনুসন্ধান করতে সক্ষম।
6. Data Integrity: RDBMS ডেটার সঠিকতা এবং নির্ভরযোগ্যতা নিশ্চিত করতে বিভিন্ন ডেটা Constraint (যেমন NOT NULL, UNIQUE, CHECK) প্রয়োগ করে।
7. Transaction Support: RDBMS ডেটাবেস ট্রানজেকশনগুলির জন্য পূর্ণ সমর্থন প্রদান করে, যেখানে একটি ট্রানজেকশন সম্পূর্ণভাবে সফল হলে তা গ্রহণ করা হয়, অথবা কোনো সমস্যা হলে পুরো ট্রানজেকশনটি বাতিল করা হয়।

Machine Learning এ RDBMS এর প্রয়োজন কেন?

Machine Learning (ML) এ RDBMS এর প্রয়োজনীয়তা বেশ গুরুত্বপূর্ণ, কারণ:

1. **Data Storage:** Machine Learning মডেলগুলোর জন্য প্রচুর ডেটা প্রয়োজন। RDBMS এই ডেটাগুলো সুশৃঙ্খলভাবে সংরক্ষণ করে এবং সহজে অ্যাক্সেসযোগ্য রাখে, যা মডেল ট্রেনিং এবং টেস্টিংয়ের জন্য গুরুত্বপূর্ণ।
2. **Data Retrieval and Preprocessing:** Machine Learning মডেল তৈরি করার আগে ডেটা ক্লিনিং, ট্রান্সফরমেশন, এবং ফিচার ইঞ্জিনিয়ারিং করতে হয়। SQL এর মাধ্যমে ডেটা দ্রুত এবং কার্যকরীভাবে রিপ্রিসেস করা যায়।
3. **Scalability:** RDBMS গুলি বড় আকারে ডেটা হ্যান্ডেল করতে সক্ষম, যা Machine Learning মডেলগুলোর ক্ষেত্রে এবং জটিল ডেটাসেট ব্যবহার করতে সাহায্য করে।
4. **Data Integrity & Consistency:** Machine Learning মডেলগুলির জন্য ডেটা সঠিক এবং নির্ভরযোগ্য হওয়া খুব গুরুত্বপূর্ণ। RDBMS এর ACID গুণাবলী ডেটা সঠিকতা ও কনসিস্টেন্সি নিশ্চিত করে।
5. **Integration with Other Systems:** RDBMS অন্যান্য সিস্টেমের সাথে সহজেই ইন্টিগ্রেট হতে পারে (যেমন Data Warehouse, API), যা Machine Learning মডেল তৈরি করার জন্য ডেটার বিভিন্ন উৎসকে একত্রিত করতে সহায়ক।
6. **Batch Processing:** অনেক Machine Learning অ্যালগরিদম ব্যাচ ভিত্তিক ডেটা নিয়ে কাজ করে। RDBMS গুলি বড় ডেটা অপারেশন করতে সক্ষম, যা মডেল ট্রেনিংয়ের জন্য প্রয়োজনীয়।

Machine Learning এবং RDBMS এর সম্পর্ক:

- ✚ **Data Engineering:** Machine Learning মডেল ট্রেনিংয়ের আগে ডেটা প্রিপ্রেসিং এবং ট্রান্সফরমেশন করা হয়, যা সাধারণত RDBMS এ করা হয়।
 - ✚ **Feature Engineering:** RDBMS থেকে ডেটা নিয়ে ফিচার তৈরি করা হয়, যা Machine Learning মডেলকে ট্রেন করার জন্য ব্যবহার হয়।
 - ✚ **Data Pipeline:** Machine Learning সিস্টেমগুলো সাধারণত Data Pipeline ব্যবহার করে, যেখানে RDBMS গুরুত্বপূর্ণ ভূমিকা পালন করে ডেটা ইনজেশন, ক্লিনিং এবং স্টোরেজের মাধ্যমে।
- এভাবে, RDBMS Machine Learning সিস্টেমে ডেটা সংরক্ষণ এবং প্রক্রিয়াকরণের জন্য অত্যন্ত গুরুত্বপূর্ণ, যা মডেল ট্রেনিংয়ের জন্য প্রয়োজনীয় ডেটার মান এবং নির্ভরযোগ্যতা নিশ্চিত করে।

কোম্পানির ডেটা RDBMS এ সংরক্ষিত থাকে, যা SQL এর মাধ্যমে প্রয়োজন অনুযায়ী বের করা যায়। সঠিক ডেটা অনুসন্ধান ও ব্যবস্থাপনার জন্য RDBMS এবং SQL এর জ্ঞান জানা অত্যন্ত জরুরি।

RDBMS (Relational Database Management System) এর কিছু জনপ্রিয় সফটওয়্যার নাম হলো:

MySQL, **PostgreSQL**, Oracle Database, Microsoft SQL Server, SQLite, IBM Db2, MariaDB, Amazon RDS (Relational Database Service)

এই সফটওয়্যারগুলো RDBMS এর কাঠামো ব্যবহার করে ডেটাবেস ম্যানেজমেন্ট, ডেটা সংরক্ষণ, এবং SQL এর মাধ্যমে ডেটার পরিচালনা করতে সহায়তা করে।

2. Database Normalization concepts.

ডাটাবেজ নরমালাইজেশন একটি প্রক্রিয়া যার মাধ্যমে ডাটাবেজের কাঠামোকে আরও কার্যকরী এবং সহজ পরিচালনাযোগ্য করা হয়। 1NF, 2NF, এবং 3NF তিনটি পর্যায়, যা টেবিলের তথ্যের কাঠামো এবং সম্পর্ক নির্ধারণ করে।

এখানে একটি উদাহরণ দিয়ে 1NF, 2NF, এবং 3NF এর ব্যাখ্যা দেওয়া হলো।

উদাহরণ:

ধরি, আমাদের একটি টেবিল আছে যেখানে ছাত্রদের (students) এবং তাদের কোর্সের তথ্য রয়েছে।

অপরিষ্কৃত টেবিল (Unnormalized Table):

StudentID	StudentName	Course1	Course2	Instructor1	Instructor2
1	Ratul	BANGLA	ENGLISH	Dr. Rabindranath	Dr. Donald Trump
2	Fahim	ICT	NULL	Dr. Minhaz	NULL
3	Tarif	BANGLA	MATH	Dr. Rabindranath	Dr. Farzana
4	Limon	ENGLISH	PHYSICS	Dr. Donald Trump	Dr. Einstein

1NF (প্রথম নর্মাল ফর্ম):

1NF এর মৌলিক নিয়ম হলো:

- প্রতিটি কলামে একক মান থাকতে হবে, অর্থাৎ কোনো কলামের মধ্যে একাধিক মান (Multiple values) থাকতে পারবে না।
- প্রতিটি রেকর্ড বা টুপলে (tuple) একটি নির্দিষ্ট সংখ্যা কলাম থাকতে হবে, এবং সেগুলোকে স্বতন্ত্রভাবে চিহ্নিত করা যাবে।

এই টেবিলটি 1NF তে আনার জন্য, আমাদের Course1, Course2, Instructor1, এবং Instructor2 কলামগুলোতে একাধিক মান থাকতে পারে, যা 1NF এর নিয়ম অনুযায়ী ঠিক নয়। তাই, এই কলামগুলোকে আলাদা রেকর্ডে ভাগ করতে হবে।

1NF এ রূপান্তরিত টেবিল:

StudentID	StudentName	CourseCode	Instructor
1	Ratul	BANGLA	Dr. Rabindranath
1	Ratul	ENGLISH	Dr. Donald Trump
2	Fahim	ICT	Dr. Minhaz
3	Tarif	BANGLA	Dr. Rabindranath
3	Tarif	MATH	Dr. Farzana
4	Limon	ENGLISH	Dr. Donald Trump
4	Limon	PHYSICS	Dr. Einstein

এখানে, Ratul ছাত্র দুটি কোর্স (BANGLA এবং ENGLISH) নিলেও, প্রতিটি কোর্সের জন্য আলাদা রেকর্ড আছে এবং এইভাবে মাল্টিপল ভ্যালু সরানো হয়েছে। এখন প্রতিটি কলামে একক মান রয়েছে।

2NF (দ্বিতীয় নর্মাল ফর্ম):

2NF এর নিয়ম হলো:

1. প্রথমত, টেবিলটি 1NF তে থাকতে হবে।
2. সমস্ত আংশিক নির্ভরতা (Partial Dependency) দূর করতে হবে। অর্থাৎ, যদি কোনো কলাম প্রাথমিক কীগুলির আংশিক অংশের উপর নির্ভরশীল থাকে, তাহলে সেই কলামটিকে আলাদা টেবিলের মধ্যে স্থানান্তর করতে হবে।

এখানে, StudentID এবং CourseCode এর সমন্বয়ে প্রাথমিক কী তৈরি হয়েছে। তবে, Instructor শুধুমাত্র CourseCode এর উপর নির্ভরশীল, এটি StudentID এর উপর নয়। সুতরাং, Instructor সম্পর্কিত তথ্যকে আলাদা টেবিলে রাখতে হবে।

2NF এ রূপান্তরিত টেবিল:

Student_Course টেবিল:

StudentID	StudentName	CourseCode
1	Ratul	BANGLA
1	Ratul	ENGLISH
2	Fahim	ICT
3	Tarif	BANGLA
3	Tarif	MATH
4	Limon	ENGLISH
4	Limon	PHYSICS

Course টেবিল:

CourseCode	Instructor
BANGLA	Dr. Rabindranath
ENGLISH	Dr. Donald Trump
ICT	Dr. Minhaz
MATH	Dr. Farzana
PHYSICS	Dr. Einstein

এখানে, Student_Course টেবিলটি ছাত্রদের এবং কোর্সগুলির সম্পর্ক ধারণ করে এবং Course টেবিলটি কোর্স এবং তাদের ইনস্ট্রাক্টরের সম্পর্ক ধারণ করছে।

3NF (তৃতীয় নর্মাল ফর্ম):

3NF (তৃতীয় নর্মাল ফর্ম) এ রূপান্তর করার জন্য, আমাদের লক্ষ্য হবে Transitive Dependency (অপ্রত্যক্ষ নির্ভরতা) দূর করা। 3NF তে পৌঁছানোর জন্য দুটি শর্ত পূর্ণ করতে হবে:

1. টেবিলটি 2NF তে থাকতে হবে।
2. কোন ক্ষেত্রের মধ্যে Transitive Dependency (অপ্রত্যক্ষ নির্ভরতা) থাকা উচিত না। অর্থাৎ, কোন কলাম যদি অন্য কলামের মাধ্যমে প্রাথমিক কীগুলির উপর নির্ভরশীল হয়, তবে তাকে আলাদা টেবিলের মধ্যে নিয়ে আসা উচিত।

এখানে Student_Course টেবিলের মধ্যে StudentName কলামটি StudentID এর উপর নির্ভরশীল। কিন্তু, StudentName সরাসরি কোর্স বা ইনস্ট্রাক্টরের সাথে সম্পর্কিত নয়, বরং StudentID এর মাধ্যমে সংযুক্ত। সুতরাং, StudentName কে আলাদা একটি Student টেবিলে স্থানান্তর করতে হবে।

এখন, Course টেবিলটি Instructor এর নাম এবং কোর্স কোড ধারণ করছে, যা ইতিমধ্যে CourseCode এর উপর নির্ভরশীল এবং এখানে কোনও অপ্রত্যক্ষ নির্ভরতা নেই।

3NF রূপান্তরিত টেবিল:

Student টেবিল:

StudentID	StudentName
1	Ratul
2	Fahim
3	Tarif
4	Limon

Student_Course টেবিল:

StudentID	CourseCode
1	BANGLA
1	ENGLISH
2	ICT
3	BANGLA
3	MATH
4	ENGLISH
4	PHYSICS

Course টেবিল:

CourseCode	Instructor
BANGLA	Dr. Rabindranath
ENGLISH	Dr. Donald Trump
ICT	Dr. Minhaz
MATH	Dr. Farzana
PHYSICS	Dr. Einstein

এখন:

1. Student টেবিলটি ছাত্রদের নাম এবং আইডি ধারণ করছে।
2. Student_Course টেবিলটি ছাত্রদের কোর্সের সম্পর্ক ধারণ করছে।
3. Course টেবিলটি কোর্স এবং ইনস্ট্রাক্টরের সম্পর্ক ধারণ করছে।

এভাবে, 3NF তে রূপান্তর করার মাধ্যমে, StudentName এর অপ্রত্যক্ষ নির্ভরতাকে দূর করা হয়েছে এবং ডাটাবেজের কাঠামো আরও পরিষ্কার এবং আরও কার্যকরী হয়েছে।

প্রাকৃতিক ভাষা প্রক্রিয়াকরণ (NLP) এ ডাটাবেস নরমালাইজেশন ব্যবহৃত হয় যাতে টেক্সট ডাটাকে সঠিকভাবে এবং কার্যকরভাবে সংরক্ষণ করা যায়। যখন আমরা টেক্সট ডাটার সাথে সম্পর্কিত তথ্য সংরক্ষণ করি, তখন ডুপ্লিকেট তথ্য কমানোর জন্য ডাটাবেস নরমালাইজেশন গুরুত্বপূর্ণ। উদাহরণস্বরূপ, শব্দের বিভিন্ন রূপ ("run", "running") একত্রিত করে শুধুমাত্র মূল রূপ ("run") রাখা হয়। এছাড়া, এনটিটি রিকগনিশন (NER) এর মাধ্যমে ব্যক্তি, স্থান বা সংগঠন সংক্রান্ত তথ্য আলাদা টেবিলে রাখা হয়। এইভাবে, টেবিলগুলির মধ্যে সম্পর্ক স্পষ্ট থাকে এবং ডাটাবেসে অপ্রয়োজনীয় পুনরাবৃত্তি কমে যায়। এতে ডাটাবেস আরও কার্যকরী ও দ্রুত হয়, যা NLP কাজের গতি বাঢ়ায়।

3. Introduction to NoSQL databases.

NoSQL ডেটাবেস হলো এমন একটি ডেটাবেস যা Relational Databases (RDBMS) এর বাইরে কাজ করে এবং ফ্লেক্সিবল, স্কেলেবল ডেটা স্টোরেজের জন্য ডিজাইন করা হয়। এর মাধ্যমে বড় পরিমাণ এবং বিভিন্ন ধরনের ডেটা সংরক্ষণ করা সম্ভব, যেমন JSON, XML, বা ডকুমেন্ট ফর্ম্যাটে। NoSQL ডেটাবেসে ডেটা টেবিলের মতো কাঠামোতে রাখা হয় না, বরং কী-ভ্যালু, ডকুমেন্ট, কলাম ফ্যামিলি, বা গ্রাফ ফর্ম্যাটে সংরক্ষিত থাকে।

এগুলি দ্রুত পারফরমেন্স, সহজ ক্ষেপিং এবং উচ্চ কার্যকারিতা প্রদান করে, বিশেষ করে যখন ডেটার পরিমাণ বড় এবং দ্রুত পরিবর্তিত হয়। উদাহরণস্বরূপ, MongoDB, Cassandra, Redis ইত্যাদি জনপ্রিয় NoSQL ডেটাবেস।

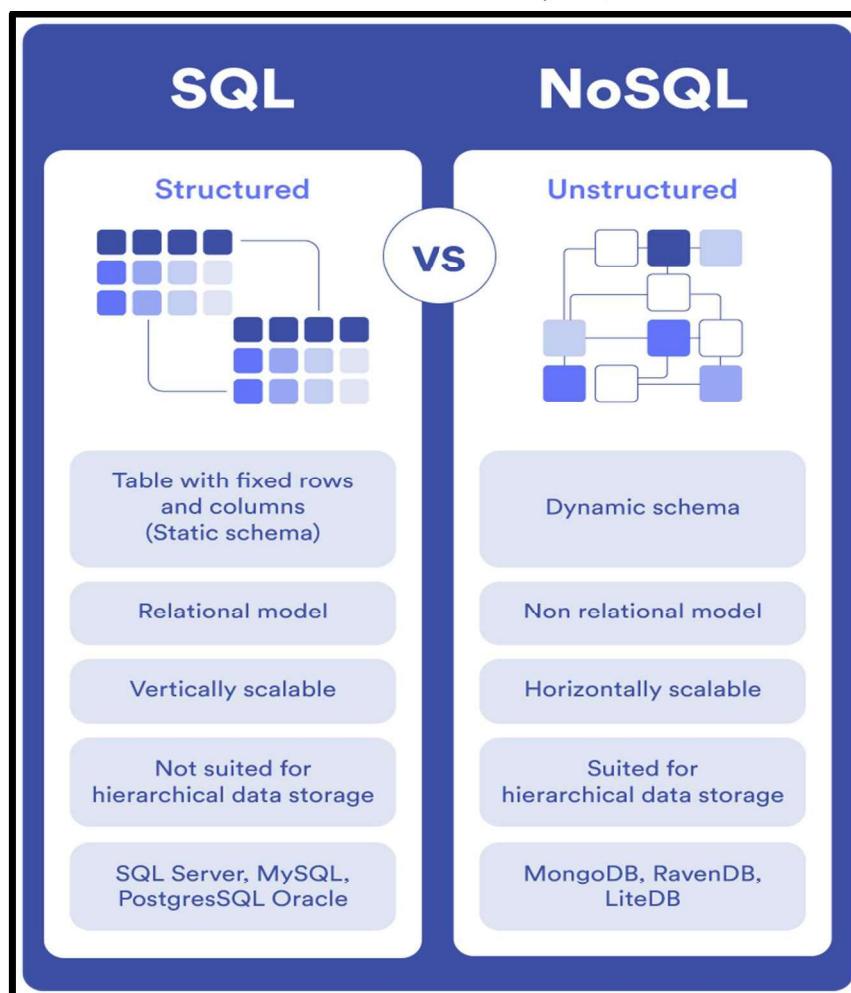
মূল সুবিধা:

1. নমনীয়তা: বিভিন্ন ধরনের ডেটা সংরক্ষণ করতে পারে।
2. ক্ষেপেবিলিটি: সহজে বড় পরিমাণ ডেটা হ্যান্ডল করা যায়।
3. হাই পারফরমেন্স: দ্রুত ডেটা প্রক্রিয়াকরণ সক্ষম।

NoSQL ডেটাবেস মূলত সোশ্যাল মিডিয়া, ই-কমার্স, IoT এবং অন্যান্য বড় প্ল্যাটফর্মে ব্যবহৃত হয়।

ফেসবুকে পোস্ট দেওয়ার সময় আপনি বিভিন্ন ধরনের কন্টেন্ট আপলোড করতে পারেন—যেমন টেক্সট, ছবি, এবং ভিডিও। তাকে শুধুমাত্র একটা ক্যাটাগরিতে ফেলা যায় না। ফেসবুক তার নিজস্বভাবে এই কন্টেন্টগুলোকে বিভিন্ন ফরম্যাটে গুছিয়ে রাখে এবং পরিবেশন করে। এটি করা হয় NoSQL ডাটাবেসের মাধ্যমে। NoSQL ডাটাবেসে ডেটা সাধারণত বিভিন্ন ধরনের ডকুমেন্ট, গ্রাফ বা ট্রি আকারে সংরক্ষিত থাকে, যেখানে সম্পর্কিত ডেটা একত্রে রাখা হয়, তবে এটি সম্পর্কিত টেবিল বা ছকে সংরক্ষিত হয় না, যেমন SQL ডাটাবেসে থাকে।

তবে, যখন আমাদের মেশিন লার্নিং (Machine Learning) এর জন্য ডেটা ব্যবহারের প্রয়োজন হয়, তখন বেশিরভাগ ক্ষেত্রেই NoSQL থেকে SQL (Structured Query Language) ডাটাবেসে ডেটা স্থানান্তর করতে হয়। কারণ SQL ডাটাবেসে ডেটা টেবিল বা ছক আকারে থাকে, যা মেশিন লার্নিং অ্যালগরিদমের জন্য প্রক্রিয়াকরণ এবং বিশ্লেষণ করতে সুবিধাজনক। SQL ডাটাবেসে ডেটা সম্পর্কিত এবং কাঠামোবদ্ধ থাকে, তাই এর মাধ্যমে ডেটার বিশ্লেষণ সহজ হয়।



4. Selecting data with SQL statements.

DataFrame:

DataFrame/Table/ছক হলো একটি টেবিলের মতো ডাটা স্ট্রাকচার, যা সাধারণত পান্ডাস (Pandas) লাইব্রেরিতে ব্যবহৃত হয়। এটি দুটি দিক দিয়ে সংগঠিত থাকে: রো (rows) এবং কলাম (columns)। সহজভাবে, এটি একটি দ্বিমাত্রিক টেবিল যেখানে বিভিন্ন ধরনের ডেটা যেমন সংখ্যা, স্ট্রিং বা তারিখ রাখা যায়।

DataFrame এ:

- কলাম হলো একেকটি বৈশিষ্ট্য বা ক্যাটেগরি (যেমন, নাম, বয়স, শহর)।
- রো হলো একেকটি ডেটা রেকর্ড (যেমন, একজন ব্যক্তির নাম, বয়স এবং শহর)।

এটি স্প্রেডশীট বা টেবিলের মতো হয়, যেখানে আমরা ডেটা সহজে দেখতে, সংরক্ষণ করতে এবং বিশ্লেষণ করতে পারি।

উদাহরণ:

ধরা যাক, একটি DataFrame/Table/ছকে এ তিনটি ব্যক্তির নাম, বয়স এবং শহর রাখা হয়েছে:

Name	Age	City
Alice	25	New York
Bob	30	Los Angeles
Charlie	35	Chicago

এখানে, Name, Age, এবং City কলাম এবং 1, 2, 3 হলো রো ইনডেক্স। DataFrame এ ডেটা টেবিলের মতো আকারে থাকে, যা বিশ্লেষণ বা প্রক্রিয়াকরণের জন্য খুবই উপকারী।

RDBMS এর বহুল জনপ্রিয় সফটওয়্যার নাম হলো PostgreSQL. যা নিম্নোক্ত সাইট থেকে download করা যায়ঃ

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

SQL একটি ডেটাবেস Query Language, যা পাইথনের মতো প্রোগ্রামিং ভাষা হলেও, ডেটাবেস পরিচালনার জন্য বিশেষভাবে ডিজাইন করা। পাইথনের তুলনায় SQL কমপ্লেক্স নয়, সরলভাবে ডেটা খোঁজা, যুক্ত করা, আপডেট করা বা মুছে ফেলার কাজ করে।

RDBMS (Relational Database Management System) একরকমভাবে Excel এর মত কাজ করে। যেমন, Excel ফাইলে আমরা অনেক শীট (Sheet) তৈরি করতে পারি, তেমনি RDBMS এ আমরা বিভিন্ন টেবিল (Table) তৈরি করতে পারি।

এখানে Excel এর দুটি শীট (Sheet1 ও Sheet2) এবং তাদের শিরোনাম এবং ডেটা দিয়ে উদাহরণ দেয়া হলো:

শীট ১: "Personal_Info"

এই শীটে আমরা "Name", "Age", এবং "City" শিরোনামে একটি টেবিল তৈরি করেছি। এখানে কয়েকটি তথ্য আছে:

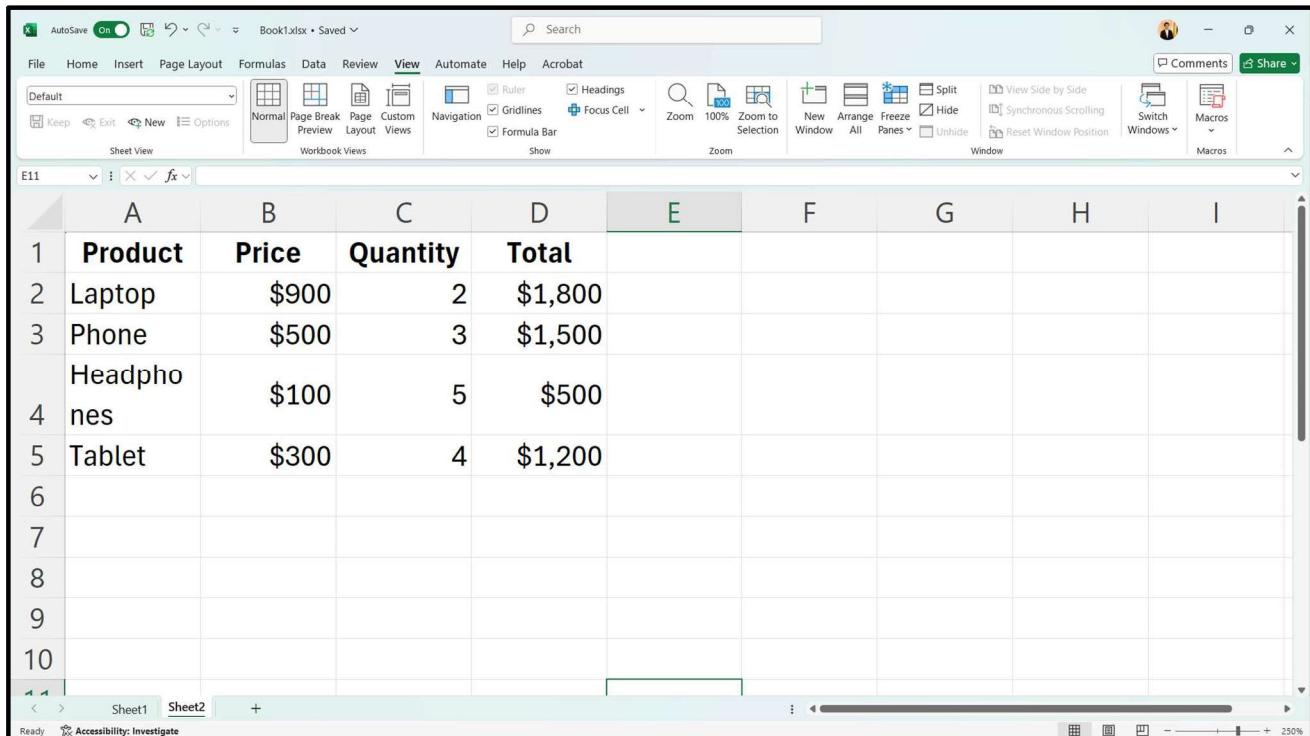
Name	Age	City
Alice	25	New York
Bob	30	Los Angeles
Charlie	22	Chicago
Diana	28	Miami

শীট ২: "Product_Info"

এই শীটে আমরা "Product", "Price", "Quantity", এবং "Total" শিরোনামে আরেকটি টেবিল তৈরি করেছি। এখানে কিছু প্রোডাক্টের তথ্য দেয়া হলো:

Product	Price	Quantity	Total
Laptop	\$900	2	\$1800
Phone	\$500	3	\$1500
Headphones	\$100	5	\$500
Tablet	\$300	4	\$1200

অর্থাৎ, RDBMS এ আমরা বিভিন্ন টেবিল তৈরি করে, যেমন Excel এ শীট তৈরি করা হয়।



The screenshot shows a Microsoft Excel spreadsheet titled "Book1.xlsx". The "View" tab is selected in the ribbon. A table is displayed with the following data:

	A	B	C	D	E	F	G	H	I
1	Product	Price	Quantity	Total					
2	Laptop	\$900	2	\$1,800					
3	Phone	\$500	3	\$1,500					
4	Headphones	\$100	5	\$500					
5	Tablet	\$300	4	\$1,200					
6									
7									
8									
9									
10									

The status bar at the bottom shows "Ready" and "Accessibility: Investigate".

এখানে SQL-এ ব্যবহৃত সবচেয়ে সাধারণ ডেটা টাইপগুলোর সংক্ষিপ্ত ব্যাখ্যা দেয়া হলো:

1. INT / INTEGER: পূর্ণসংখ্যা সংরক্ষণ করে। যেমন: 10, -200।
2. VARCHAR(n): ভ্যারিয়েবল দৈর্ঘ্যের স্ট্রিং (অক্ষর)। যেমন: 'Alice', 'New York'।
3. DATE: শুধু তারিখ (YYYY-MM-DD) সংরক্ষণ করে। যেমন: '2025-01-08'।
4. DATETIME: তারিখ এবং সময় উভয় (YYYY-MM-DD HH:MM:SS) সংরক্ষণ করে। যেমন: '2025-01-08 15:30:00'।
5. BOOLEAN: সত্য (TRUE) বা মিথ্যা (FALSE) মান সংরক্ষণ করে। যেমন: TRUE, FALSE।
6. DECIMAL / NUMERIC: সঠিক দশমিক সংখ্যা সংরক্ষণ করে। যেমন: 12.45, 500.75।
7. CHAR(n): স্থির দৈর্ঘ্যের স্ট্রিং। যেমন: 'A', 'NY'।
8. FLOAT: ভাসমান বিন্দু সংখ্যা সংরক্ষণ করে। যেমন: 3.14, 2.718।
9. TEXT: বড় আকারের টেক্স্ট বা স্ট্রিং সংরক্ষণ করে। যেমন: 'This is a long description.'

এই ডেটা টাইপগুলো বিভিন্ন প্রকার ডেটা সংরক্ষণের জন্য ব্যবহৃত হয়।

PostgreSQL ইনস্টল করার মাধ্যমে একটি RDBMS প্ল্যাটফর্ম তৈরি হয়, যেখানে আপনি Database বা excel ফাইল তৈরি করতে পারেন।

Excel এর মধ্যে অনেক sheet তৈরি করা যায়, তেমনি একটি database-এ ও অনেক Table থাকে পারে। একটি database বিভিন্ন Table-এর মাধ্যমে গঠন করা হয়, যেখানে প্রতিটি Table ডেটা সংরক্ষণ করে।

এটি একটি SQL CREATE TABLE স্টেটমেন্ট, যা একটি নতুন টেবিল তৈরি করার জন্য ব্যবহৃত হয়। এখানে একটি customers নামের টেবিল তৈরি করা হচ্ছে এবং এর বিভিন্ন কলামের (columns) সংজ্ঞা দেওয়া হচ্ছে।

```
create table customers
(
    "id" int primary key,
    "name" varchar(50) not null,
    "age" int not null,
    "city" char(50),
    "salary" numeric
);
```

CREATE TABLE customers

এই স্টেটমেন্টটি customers নামক একটি নতুন টেবিল তৈরি করে। customers হচ্ছে টেবিলের নাম।

কলামগুলি:

1. id int primary key:

- ⊕ এখানে একটি কলাম id তৈরি করা হচ্ছে যা int (integer) ধরনের।
- ⊕ primary key নির্দেশ করছে যে এই কলামটি ইউনিক (অনন্য) হবে এবং এটি টেবিলের প্রতিটি রেকর্ডকে আলাদা আলাদা সনাক্ত করতে সাহায্য করবে।
- ⊕ primary key স্বয়ংক্রিয়ভাবে NOT NULL তৈরি করে, অর্থাৎ এই কলামে কোন মান ফাঁকা (NULL) থাকতে পারবে না।

2. name varchar(50) not null:

- ⊕ name কলামটি varchar(50) ধরনের, যা একটি স্ট্রিং (টেক্স্ট) ডেটা ধরনের কলাম এবং এতে সর্বাধিক ৫০টি চরিত্র (অক্ষর) থাকতে পারবে।
- ⊕ not null নির্দেশ করছে যে এই কলামে অবশ্যই একটি মান থাকতে হবে, ফাঁকা (NULL) মান দেওয়া যাবে না।

3. age int not null:

- ⊕ age কলামটি int ধরনের, যা একটি পূর্ণসংখ্যা (integer) ধারণ করে এবং এখানে not null ব্যবহৃত হচ্ছে, অর্থাৎ এই কলামে কোন মান ফাঁকা (NULL) থাকতে পারবে না।

4. city char(50):

- ⊕ city কলামটি char(50) ধরনের, যেখানে একটি নির্দিষ্ট দৈর্ঘ্যের (এখানে ৫০টি চরিত্র) স্ট্রিং ধারণ করা যাবে।
- ⊕ এখানে not null উল্লেখ করা হয়নি, অর্থাৎ এই কলামে ফাঁকা মান (NULL) থাকতে পারে।

5. salary numeric:

- ⊕ salary কলামটি numeric ধরনের, যা ডেসিমেল সংখ্যাকে ধারণ করতে পারে (যেমন: 1234.56)। এই ধরনের কলাম সাধারণত অক্ষ, টাকা ইত্যাদির জন্য ব্যবহার করা হয়।
- ⊕ এখানে not null উল্লেখ করা হয়নি, তাই এই কলামে ফাঁকা (NULL) মান থাকতে পারে।

সারাংশ:

এই SQL স্টেটমেন্টটি একটি customers নামক টেবিল তৈরি করছে, যার মধ্যে পাঁচটি কলাম রয়েছে: id, name, age, city, এবং salary। id একটি প্রাইমারি কী হিসেবে কাজ করবে, যা প্রতিটি গ্রাহকের জন্য ইউনিক সনাক্তকারী হবে। name এবং age কলামগুলির জন্য not null নির্ধারিত হয়েছে, তাই এগুলোর মান অবশ্যই প্রদান করতে হবে, যখন city এবং salary কলামে ফাঁকা মান রাখা যেতে পারে।

Primary কি বলতে কি বুঝা ?? আমাকে লিখে জানাও।

customers table এর মান দেখার জন্য query:

```
select *from customers
```

SELECT * FROM customers; SQL স্টেটমেন্টটি customers টেবিল থেকে সব রেকর্ড এবং সব কলাম নির্বাচন/select করে প্রদর্শন করবে। এখানে, * দ্বারা টেবিলের সব কলাম (যেমন: id, name, age, city, salary) বোঝানো হয়েছে, এবং FROM এর বাংলা অর্থ হতে। আর, customers অংশটি টেবিলের নাম নির্দেশ করে।

এই SQL স্টেটমেন্টটি customers টেবিলে তিনটি নতুন রেকর্ড ইনসার্ট করছে, যেখানে প্রতিটি রেকর্ডের জন্য id, name, age, city, এবং salary কলামে মান দেওয়া হয়েছে।

```
insert into customers
(id, name, age, city, salary)
values
(1, 'minhaz', 30, 'Dhaka', 1000),
(2, 'kabir', 29, 'sylhet', 2000),
(3, 'Ratul', 20, 'narayanganj', 500)
```

INSERT INTO customers (id, name, age, city, salary)

- INSERT INTO: এটি SQL কমান্ড, যা টেবিলের মধ্যে নতুন ডেটা (রেকর্ড) যোগ করতে ব্যবহৃত হয়।
- customers: এই অংশে customers হলো টেবিলের নাম, যেখানে নতুন রেকর্ড গুলো ইনসার্ট হবে।
- (id, name, age, city, salary): এখানে আমরা নির্দিষ্ট করেছি যে, কোন কলামগুলোতে ডেটা ইনসার্ট করা হবে।
- id, name, age, city, এবং salary হল সেই কলামগুলো যেখানে নতুন মান (value) যোগ করা হবে।

VALUES (1, 'minhaz', 30, 'Dhaka', 1000), (2, 'kabir', 29, 'sylhet', 2000), (3, 'Ratul', 20, 'narayanganj', 500)

- VALUES: এই কিওয়ার্ডটি নির্দিষ্ট করে যে, কোন মানগুলো (values) ইনসার্ট করা হবে।
- (1, 'minhaz', 30, 'Dhaka', 1000): প্রথম রেকর্ডটি। এখানে:
 - 1 হলো id কলামে মান,

- 'minhaz' হলো name কলামে মান,
 - 30 হলো age কলামে মান,
 - 'Dhaka' হলো city কলামে মান,
 - 1000 হলো salary কলামে মান।
- (2, 'kabir', 29, 'sylhet', 2000): দ্বিতীয় রেকর্ডটি। এটি একইভাবে দ্বিতীয় রেকর্ডের মানগুলো ইনসার্ট করবে।
- (3, 'Ratul', 20, 'narayanganj', 500): তৃতীয় রেকর্ডটি। তেমনি, এটি তৃতীয় রেকর্ডের মানগুলো ইনসার্ট করবে।

হাজিরা খাতা হচ্ছে, টেবিলের চমৎকার উদাহরণঃ

Roll/Primary Key (Not null)	Name (Not Null)	Day 01	Day 02	Day 03	Day 04	Day 05	Day 06	Day 07	Day 08
2022331001	Asif M.	P	p		p	P			
2022331002	Nahid I.	P		P		P			
2022331003	Sarjis A.	P	P	p	P	p			

একজন শিক্ষার্থীকে শুধুমাত্র শনাক্ত করা যায়, তার রোল নম্বর দিয়ে।

আমি যে Text ফাইলটি তোমাকে দিচ্ছি, তুমি প্রয়োজনীয় SQL টেবিল তৈরি ও ডেটা ইনসার্ট করার ক্ষমতা গুলি কপি করে, সঠিকভাবে PostgreSQL এ চালু করবে।

<https://www.programiz.com/sql/getting-started>