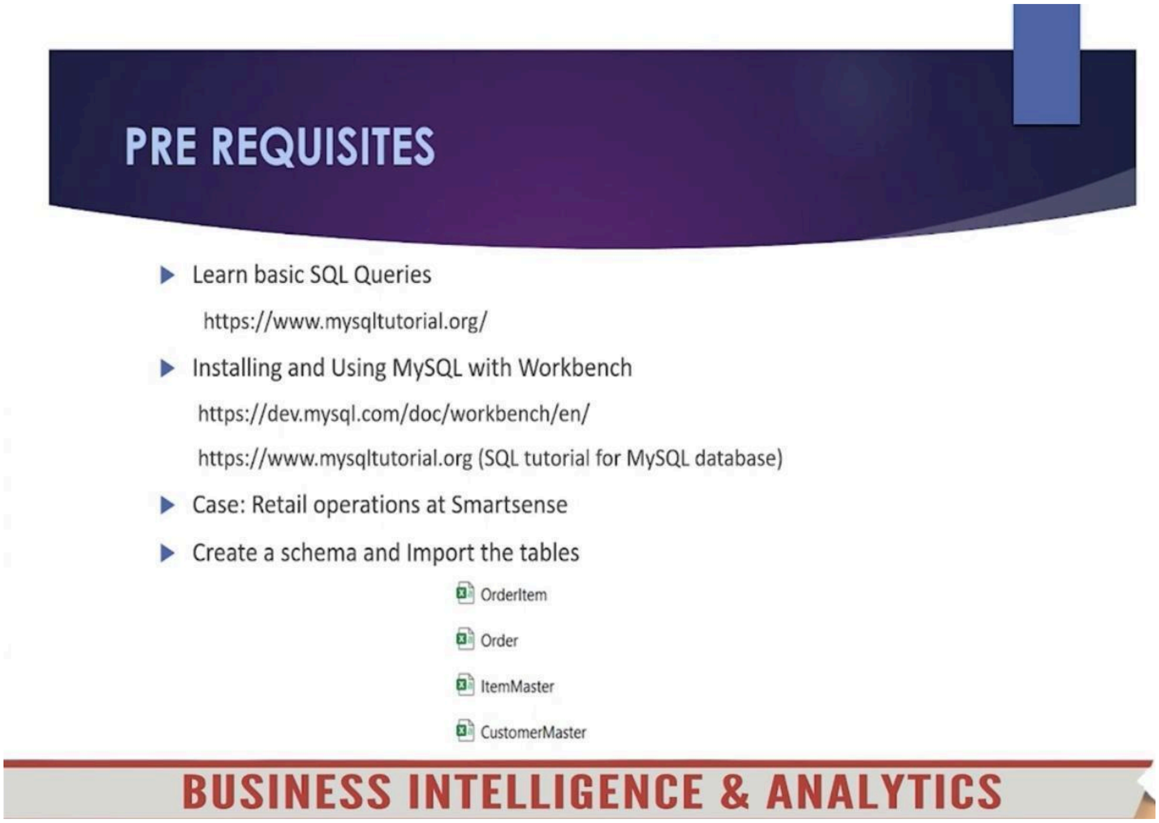
Course Name:Business Intelligence and Analytics
Professor Name:Prof. Saji.K.Mathew
Department Name:Department of Management Studies
Institute Name: Indian Institute of Technology Madras
Week:03
Lecture:09

INTRODUCTION TO SQL

Hello everyone and welcome to today's Business Intelligence and Analytics course class. Today we will be holding a session on SQL tutorial. So, I will be talking about the basics of SQL and we will also go about to see how we perform various commands in MySQL Workbench.

So, I will briefly introduce myself. I am K. R.Subisha. I am the teaching assistant for this course and also a PhD scholar in information systems in IIT Madras. So, we will go into today's lecture.



PRE REQUISITES

- Learn basic SQL Queries
  https://www.mysqltutorial.org/
- Installing and Using MySQL with Workbench
  https://dev.mysql.com/doc/workbench/en/
  https://www.mysqltutorial.org (SQL tutorial for MySQL database)
- Case: Retail operations at Smartsense
- Create a schema and Import the tables
  OrderItem
  Order
  ItemMaster
  CustomerMaster

BUSINESS INTELLIGENCE & ANALYTICS

So, before we go into this course, there are certain prerequisites for this session alone. So, it is always good if you know a small amount of basic knowledge of SQL. Even if you do not know, it is fine because we will be going through all the introduction to SQL and we will also see some basic commands or queries needed to work with SQL. So, if you have no prior knowledge on learning with SQL, then what you have to do is go to this link, that is https//www.mysqltutorial.org.

In that you will get to know about all the commands, how to work with SQL, various kind of languages like data manipulation language, data definition language, data control language and so on. So, in this session, one of the prerequisites for this session is you have to install and use MySQL. So for this course, we will be using MySQL with Workbench in order to learn SQL and work with SQL. So, how to install and all? It will be available in dev.mysql.com. You can visit the website and you will know how to install and work with it. So, it would be very helpful if you have a prior knowledge on how to work with Workbench. Even if it is difficult for you, we will address that as well in this session. Then we will be dealing with a case. So, this is not quite a prerequisite.

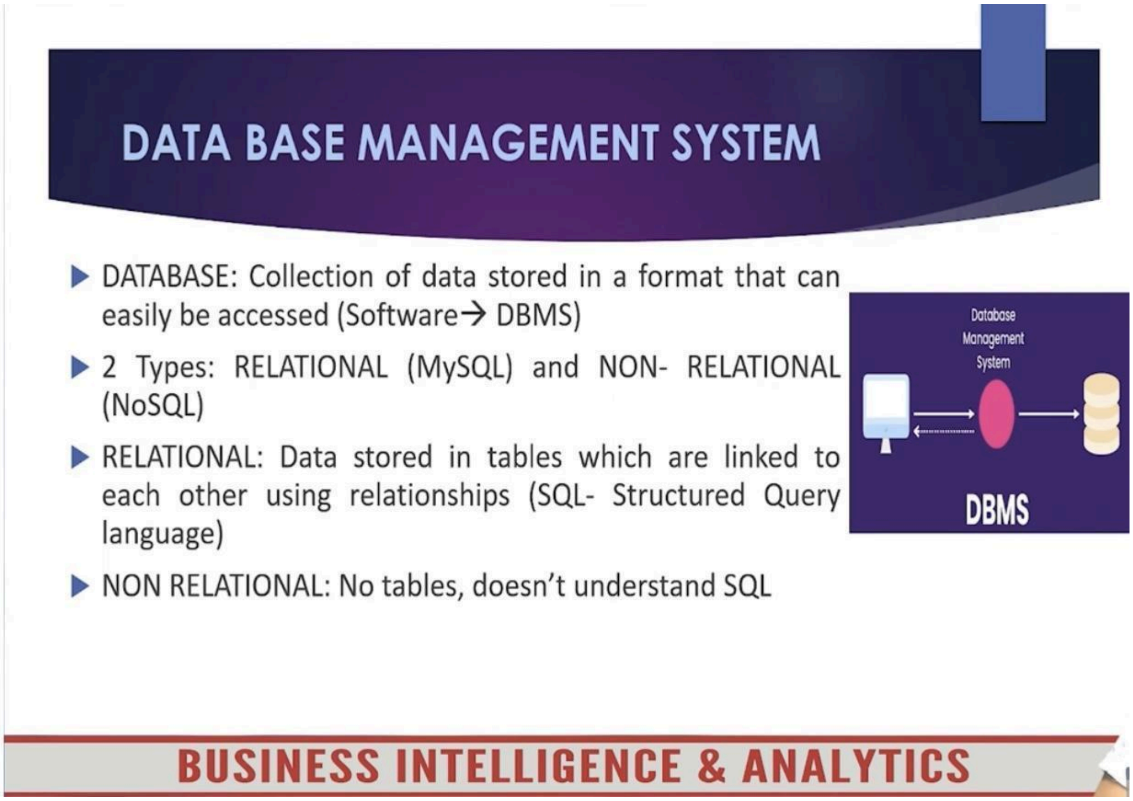
I will be dealing with this case in detail. So, we will be going through this case which is about a fictitious company called SmartSense which is a retail operations chain. So, we will be doing a bit of data analytics. I would say, there are four kinds of tables which holds the data about this SmartSense retail chain. And for this, we will be working with the MySQL Workbench, so that we can analyze what is going through in that company.

So today's agenda as I have already told, we will start with the definition of what is a DBMS. I guess you know what is the full form of DBMS. It is nothing but database management system. So, and we will be briefly going through the introduction to structured query language as you call SQL or SQL. So, it was previously called as SEQUEL because it stood for structured English query language or simply you can call it as SQL as well.

So, we will be seeing a very short session on the various commands or queries that are required to work with SQL. But if you know well in advance, it is an added advantage. And we will go through the concept of keys. For example, there are many keys as you know like primary key, foreign key, candidate key and so on and so forth. So, we will be going through that.

Then we will discuss the concept of normalization which is a very important concept as far as DBMS is concerned. So, we will be going into the concept of normalization in detail. We will be learning 1NF, 2NF, 3NF, BCNF and so on. Then we will be discussing the case which is case on Shopsense retail as I have already said. So, we will do that along with the help of MySQL with Workbench.

So, database management system. Before going into database management system, I want you to think what is a database. Yes, you guessed it right. It is nothing but a collection of data. So, where the storage, data is stored, that is called the database.



DATA BASE MANAGEMENT SYSTEM

- DATABASE: Collection of data stored in a format that can easily be accessed (Software→ DBMS)
- 2 Types: RELATIONAL (MySQL) and NON- RELATIONAL (NoSQL)
- RELATIONAL: Data stored in tables which are linked to each other using relationships (SQL- Structured Query language)
- NON RELATIONAL: No tables, doesn't understand SQL

BUSINESS INTELLIGENCE & ANALYTICS

So, data can be of many forms as you know. Data can be stored in the form of tables. Data can be stored in the form of documents or graphs. So, as far as this course is concerned, as far as the purview of this course is concerned, we are dealing with only relational database or in other words, data that is stored in the form of tables or relations. That is what we are going to learn in this course.

So, since data is stored in the form of database, there needs to be a software which will help us manage this data. What is that software called? That is what is known as DBMS or database management system. So, DBMS is basically a software that helps us manage the database, helps us manage in what sense-it can be for data storage, it can be for data retrieval, it can be to perform many actions on the data like data manipulations by using various commands like modify, alter, drop table and so on. So, as far as this course is concerned, we are dealing only with relational database or data that is stored in the form of tabular form.

So as I have already said, there are two types of data which is, or rather two types of SQL which is relational database and non-relational database. So as I already told, MySQL is used only to manage the relational database and for managing the non-relational database, there are other forms of DBMS which are no SQL types. For example, there is the example of MongoDB. So if you are interested in learning further about this, you can just go and search about non-relational databases. So, relational databases are those databases in which data is stored in the form of tables which are linked with each other using relationships.

So if you ask me, is it just one table that stores the entire database? It can be, but it necessarily need not be, because there can be multiple tables. You will come to know as we go forward because in the concepts of normalization, you will come to know why there are multiple tables needed in order to store data in the form of databases. So, it is not always one table that stores the entire database. It can be multiple tables. So, if there are multiple tables, then all these tables have to be linked to each other using relationships.

That is how the SQL works. So, in non-relational databases, as I already told you, there are no tables and it does not understand the languages or language of SQL. So, SQL is only for relational database. So, in this figure as you see, there is a system that we work with and there is a DBMS and there is a database. So, this DBMS is acting as an intermediary between the user as well as the database.

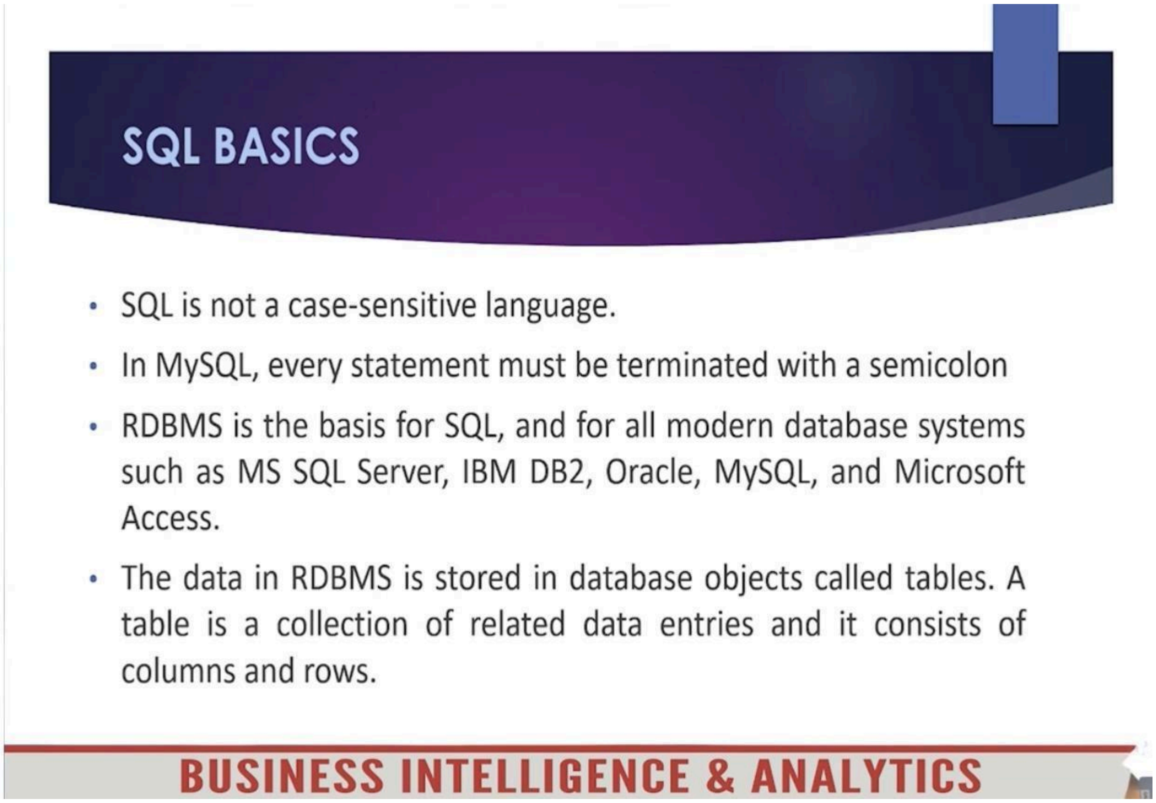
So, it is basically a software. I guess ACID properties you might have learned earlier, but I will just rush through what it stands for because we cannot discuss DBMS without knowing what ACID properties are. So, ACID stands for atomicity, consistency, isolation and durability. We will see one by one. Atomicity ensures that a transaction is treated as a single indivisible, I will start it again.

First one is atomicity. Atomicity ensures that a transaction is treated as a single indivisible unit of work. It means that either all the changes made by the transaction are applied or none of them are. For example, you are using your GPay app and you know, suppose there is a net failure, internet failure or something and the transaction got aborted. So, either the transaction should be successful or it should be non-successful or it should be failed. So, it is either a 0 or 1 and no state in between. So, that is what is known as atomicity.

The second property is consistency. So, what is consistency? Enforcing consistency ensures that if a database enters into an illegal state, that is if a violation of data integrity constraint occurs, then the process will be aborted and the changes will be rolled back to the previous legal state. So, this is basically something to ensure that no illegal things are being performed. For example, in a database of student records, if there is a particular transaction which is trying to insert a new record illegally with a duplicate of a particular student ID, so student ID should be unique. So, if there is a new record with a duplicate of a student ID, then DBMS should immediately reject that transaction in order to maintain it in a consistent state.

What is the third property? It is isolation. So, isolation ensures that the concurrent execution of multiple transaction does not result in interference or unexpected outcomes. Each transaction should be isolated from others until it is completed. For example, you are booking a flight.So, there is only one last available seat on a flight and two users are concurrently booking that. So, the transaction, if it is isolated, it should ensure that only one person is able to get that seat. So, that is what isolation means. That is two users booking, it does not interfere with each other.

Then there is durability which is the last property. So, durability guarantees that once a transaction is committed, it effects or changes to the database that are made are permanent and survive any subsequent system failures. Suppose I am teaching you and I am making some changes in this PPT. So, even if the power shutdown happens or the laptop shuts down due to some cause, then the changes made should be permanent. That is how a database works. For example, after a customer places an order and receives an order confirmation, the order details should be durably stored in a database.



SQL BASICS

- SQL is not a case-sensitive language.
- In MySQL, every statement must be terminated with a semicolon
- RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
- The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

BUSINESS INTELLIGENCE & ANALYTICS

So, a database basically has to follow all these properties which are also called as ACID properties. So, as we already talked about SQL, it stands for structured query language. And it is not a case sensitive language. So, many of you might be working with C, CPP, Python, etc.So, all those are case sensitive. And as far as SQL is concerned, it is a case insensitive language. So, you can either type in uppercase, lowercase, it does not matter. And in MySQL, every statement should end with a semicolon. So, that is the, semicolon determines the end of each query that you are typing. So, as we already talked about RDBMS, which is relational DBMS, it is the basis for SQL.

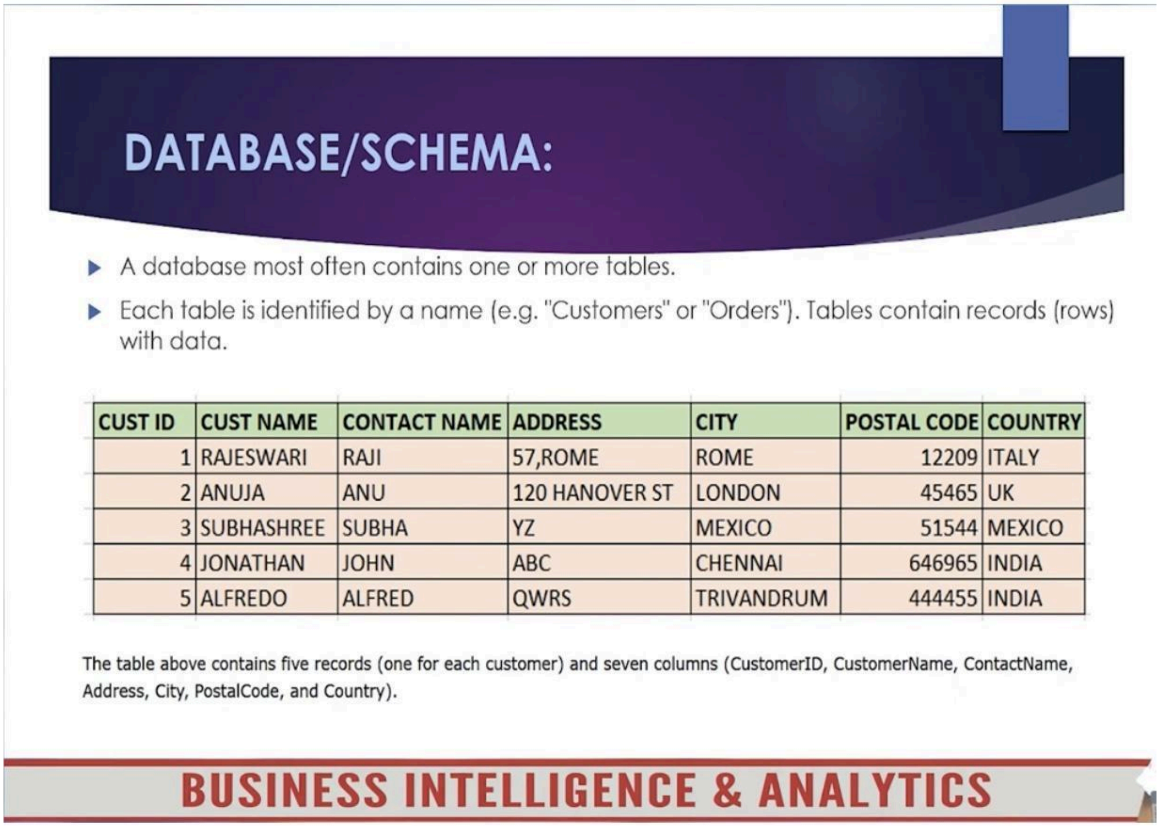
And for all the modern database systems such as MS, SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access etc. So, this also we discussed that in relational DBMS, how are the data stored? Exactly. The data are stored in the form of tables. What is a table? So, a table is a collection of related data entries and it consists of rows and columns. So just a difference, we do not call it as row and column in SQL, we call it as fields and records.

So, every table is broken up into smaller entity known as fields. So, basic fields are nothing but column names. So, say there is a customer table and the examples of the fields or the columns are customer ID, customer name, contact name, address, city, postal code, country, etc. So, all these are known as fields. So, henceforth, we will refer to rows and columns as fields and records.

So, a field is a column in a table that is designed to maintain specific information about every record in the table. So, what is a record? Yes, record is nothing but a row. And it is a row which determines the individual entry that exists in a table. Record is a horizontal entry, entity and column or a field is a vertical entity.

So, database or schema. So, what is a database? It is a structure in which the data is stored. So, database may contain one or more tables. Each table can be identified by a name. So, it is known as table name. So for example, it can be a customer table or order table.

Tables will contain records or rows with data. For example, see this database or table that we have. This is a customer table which contains fields which are customer ID, customer name, contact name and so on. And there are rows which are the horizontal entities that are mentioned. So, there are 5 records for each customer and 7 columns that are the ones mentioned in the green. In SQL, there are many commands that are used for various purposes.



DATABASE/SCHEMA:

- A database most often contains one or more tables.
- Each table is identified by a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.

CUST ID	CUST NAME	CONTACT NAME	ADDRESS	CITY	POSTAL CODE	COUNTRY
1	RAJESWARI	RAJI	57,ROME	ROME	12209	ITALY
2	ANUJA	ANU	120 HANOVER ST	LONDON	45465	UK
3	SUBHASHREE	SUBHA	YZ	MEXICO	51544	MEXICO
4	JONATHAN	JOHN	ABC	CHENNAI	646965	INDIA
5	ALFREDO	ALFRED	QWRS	TRIVANDRUM	444455	INDIA

The table above contains five records (one for each customer) and seven columns (CustomerID, CustomerName, ContactName, Address, City, PostalCode, and Country).

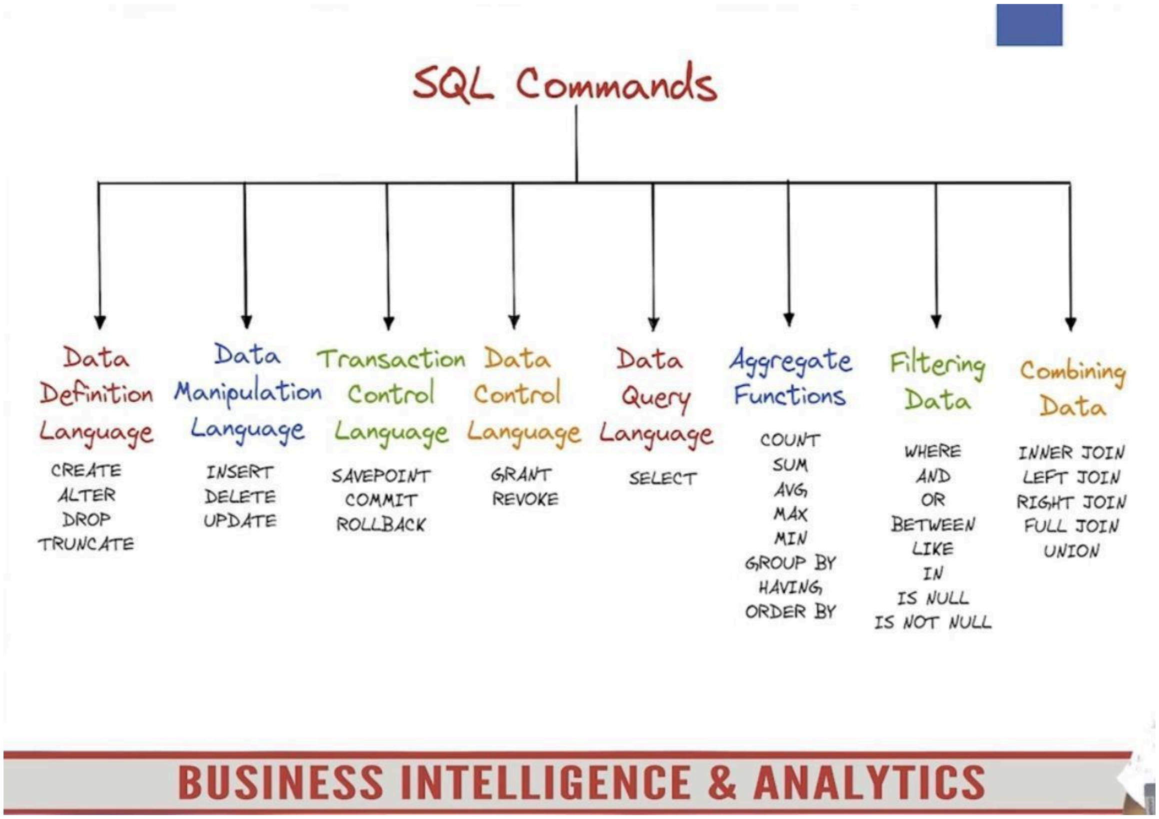
BUSINESS INTELLIGENCE & ANALYTICS

For example, it can be for defining something which come under data definition languages. So all these commands, we have divided into multiple languages. The most important ones are data definition language, data manipulation language, transaction control language, data control language and data query language and so on. We will go into one by one and not very much in detail because it will take a lot of time. So, I urge you to go back and read about all these queries and have a basic understanding of what these are, so that it will be easy while working with an actual problem.

So, what are data definition languages? These define what we want to do, what we want to work with. For example, suppose if we want to create a table, so we are defining in order to create a table. That is why this create command is used. So, create, alter, drop, truncate etc are forms of data definition languages. So, in data manipulation language, if we suppose if we want to modify a record or a field in a particular table, then we use insert or delete or update.So, all these are part of data manipulation languages.

So, the major things that we will be working will be mostly data manipulation and data querying. And all the functions that I mentioned towards the last three parts, which are aggregation functions or aggregate functions, these do mathematical operations or you know, like if we want to get a count of how many records are present in a particular column, then we use the count function. So, there are multiple functions like sum, average, max, min, group by, order by etc.

So for filtering data, that is another part of SQL where you want to filter a particular section of a data. For that you can use clauses. For example, where clause, say there is an example, select customer name from a particular table where customer country is India. So, we are giving a filtering option for this, wherein we are filtering the customers who are part of India. And the last one, but the most important one, it is combining data. As I already told, there may be multiple tables. So, it is not always the case that we work with a single table.

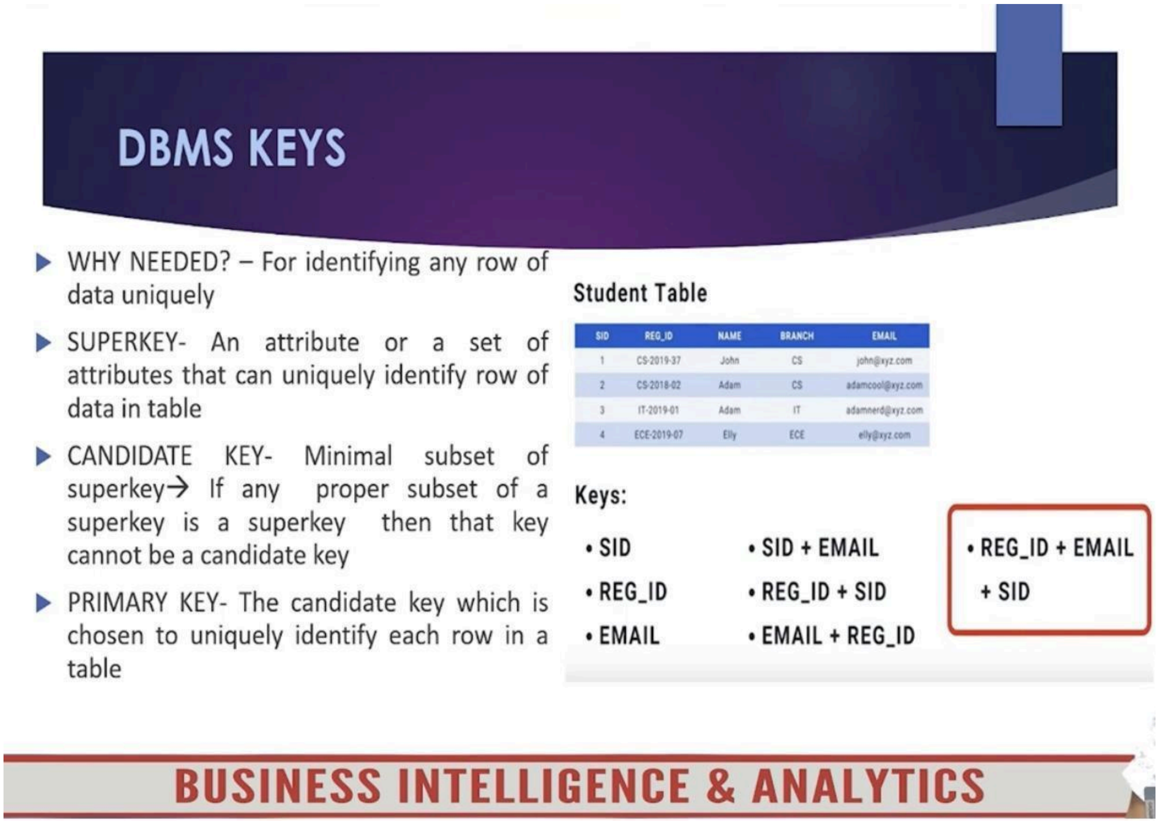


If we are working with a single table, it will be very easy. You know, there will be no complex queries. But if we are working with multiple tables, say more than two tables, we have to compare and contrast and get some analysis out of two tables, then what we have to do is, we have to perform combining operations between two tables, so that we can work with the queries. So, some of the combining operations are inner join, left join, right join, full join etc. So, all this we will see in detail.

The next topic is DBMS keys. So, what are keys? Key is something that is used for identifying any row of our table or data uniquely. So, there are multiple, you know, types of keys. First, we will be discussing about super key. For that, I need you to, you know, see this table or this student table that I have, you know, that is in front of us. It has the fields which are student ID, registration ID, name, branch and email.

So, there are five fields and four records. So, what is a super key? A super key is an attribute or a set of attributes that can uniquely identify a particular row of data in a table. For example, if we want to uniquely identify this, you know, uniquely identify a row in this table, then what are the attributes that can, that we can combine in order to do that? See, registration ID will be unique, right? Email, that is, that can be unique. So, we can select email and also student ID. So, suppose we take all these three attributes and combine it together. Will it be able to identify a particular row uniquely? Yes, right.

So, suppose we say student ID is 1, registration ID is CS 2019 37 and email is john@ xyz.com. So, we combine that and give it as an input, then it will return the first row, right? So, it is able to uniquely identify that row. So, that is what is known as super key.



DBMS KEYS

- WHY NEEDED? – For identifying any row of data uniquely
- SUPERKEY- An attribute or a set of attributes that can uniquely identify row of data in table
- CANDIDATE KEY- Minimal subset of superkey→ If any proper subset of a superkey is a superkey then that key cannot be a candidate key
- PRIMARY KEY- The candidate key which is chosen to uniquely identify each row in a table

Student Table

SID	REG_ID	NAME	BRANCH	EMAIL
1	CS-2019-37	John	CS	john@xyz.com
2	CS-2018-02	Adam	CS	adamcool@xyz.com
3	IT-2019-01	Adam	IT	adamnerd@xyz.com
4	ECE-2019-07	Elly	ECE	elly@xyz.com

Keys:

- SID
- REG_ID
- EMAIL
- SID + EMAIL
- REG_ID + SID
- EMAIL + REG_ID
- REG_ID + EMAIL + SID

BUSINESS INTELLIGENCE & ANALYTICS

What is a candidate key? Candidate key is a minimal subset of super key.For example, if any proper subset of a super key, then the key cannot be candidate key. What I mean to say is that, for example, it need not be, it need not be the case where we select multiple attributes   for uniquely identifying, when it can be done by a single attribute. For example, here the student ID itself can identify the table uniquely. Then why do we need three attributes? Instead, we can use a single attribute.

So, candidate key, it is a minimal subset. For example, student ID plus email, that can uniquely identify it. Registration ID plus student ID, that also can uniquely identify the rows. Email plus registration ID, that also can uniquely identify the rows. So, candidate key is a minimal subset of super key. I will ask you a question, can name uniquely identify each row in this table? No, right? Because there are two people who are named as Adam.

So, name cannot be individually used to uniquely identify any row in this table. So, that is not a primary key. So, what is a primary key? So, we have discussed what is a candidate key, which is a minimal subset. So, we are just taking those attributes that we need to uniquely identify.

So, what is a primary key? There are multiple candidate keys. For example, say student ID, it is a candidate key. Registration ID, it is a candidate key. So, you particularly would be choosing one attribute that you want as a primary key. That is what is known as primary key. So, the key that you choose to uniquely identify each row in a table is known as primary key.

For example, I can choose student ID. That is the most logical option here. So, usually it is very unique. It is unique for each student. So, thereby it can identify all the rows uniquely. So, those are the keys.

And foreign key, this is yet another concept, which is very important. So, foreign key is an attribute in a table, which is used to create a relationship of that table with another table. For example, I have already told you, there will be multiple tables. So, in this example, we have two tables. One is student table, another is branch table. So, can there be fields which repeat in both the tables that are unique, that are not unique, and those repeat in both the tables.

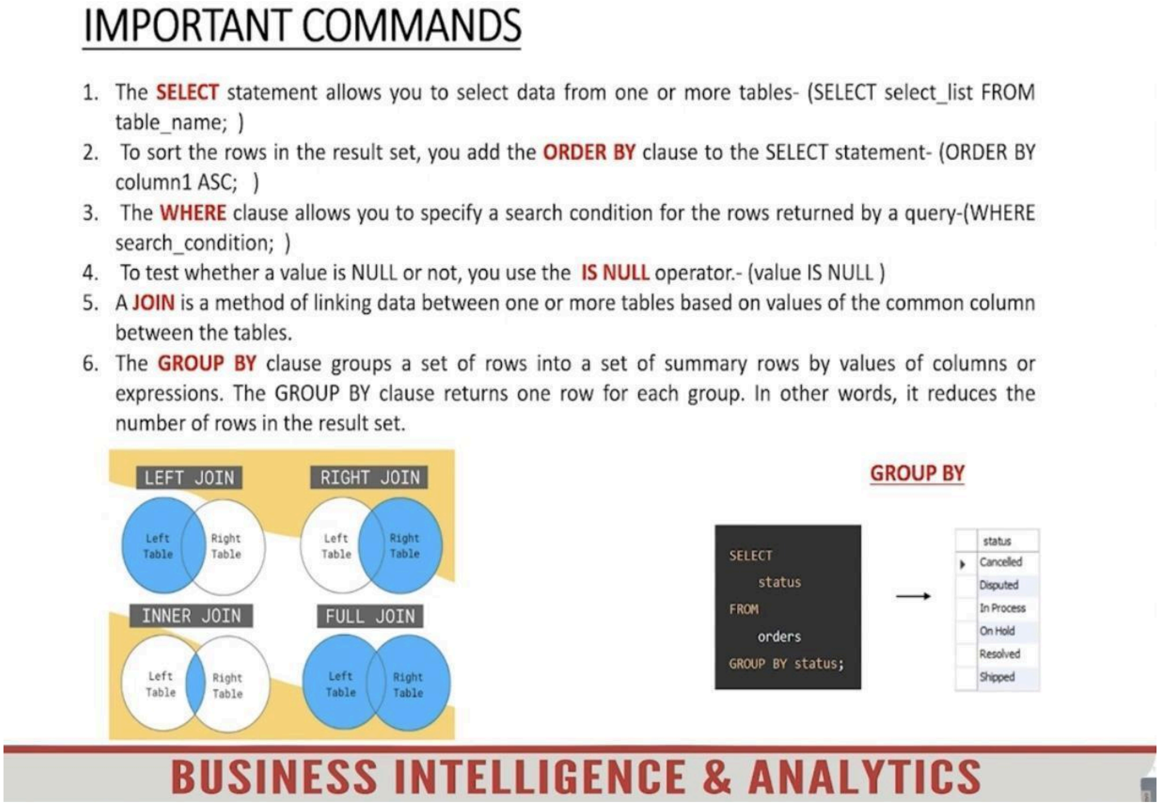
For example, see this example. There is branch code that is there in student table as well as well as branch table. So, it is not unique to one table. It repeats in multiple tables. So, the attribute in a table, which is used to create a relationship of that table with another table, that is known as foreign key. So branch code, which is a primary key in branch table, is also present in student table.

So, what we will do is, we will call that branch code in student table as the foreign key.

So, the branch code in branch table, where it is primary key, is referenced in another table. So, that will become a foreign key in student table.

Then the next one, composite key. What is a composite key? A key that has multiple attributes, not just one, but multiple attributes. We will go to the previous slide. Here registration ID plus email plus SID. How many attributes are there? There are three attributes. So, these three attributes means it is a composite key.

And compound key means it is a composite key with at least one attribute, which is a foreign key. So, if we form a composite key, which has one attribute that has branch code in it, then it becomes a compound key.



IMPORTANT COMMANDS

1. The **SELECT** statement allows you to select data from one or more tables- (SELECT select_list FROM table_name;)
2. To sort the rows in the result set, you add the **ORDER BY** clause to the SELECT statement- (ORDER BY column1 ASC;)
3. The **WHERE** clause allows you to specify a search condition for the rows returned by a query-(WHERE search_condition;)
4. To test whether a value is NULL or not, you use the **IS NULL** operator.- (value IS NULL)
5. A **JOIN** is a method of linking data between one or more tables based on values of the common column between the tables.
6. The **GROUP BY** clause groups a set of rows into a set of summary rows by values of columns or expressions. The GROUP BY clause returns one row for each group. In other words, it reduces the number of rows in the result set.

GROUP BY

```
SELECT
    status
FROM
    orders
GROUP BY status;
```

→

status
Cancelled
Disputed
In Process
On Hold
Resolved
Shipped

BUSINESS INTELLIGENCE & ANALYTICS

Now, we will see some of the important commands that are part of SQL. So, these commands, you have to know in order to work with SQL. The first and the most important command is ‘select’ command. So, the select statement, it allows you to select particular data from the tables. For example, the syntax of select statement is select column name from table name.

That is usually how the syntax is written. So, if we take this table, the branch table let us take, so it will be like select branch name from branch table. So, that is how the syntax is. The next command is ‘order by ‘command. So, to sort the rows in the result set, you add the order by command to the select statement. For example, if we write mention as order by column 1 ascending, that means column 1 will be written in ascending order.

So, that is why ‘order by’ command is used. And basically, it is doing a ‘sort’ operation. Then the next one, which does the filtering operation, that is the ‘where’ clause. So, the where clause allows you to specify a search condition for the rows, written by a query. So, let us move to the previous example, that is branch table. So, we can mention as select HOD from branch table where branch name is computer science. So, we have given a where clause here, which will filter the table for getting us the information where the branch name is only computer science.

So, the next is ‘is null’ operator. So is null, you will be knowing, it is just to test whether the value is null or not. The next operation is ‘join’. And join as I have already mentioned, it is a method of linking data between one or more tables, based on values of common columns between the tables. So, see the diagram in the left, it has 4 types of joins-left join, right join, inner join and full join.

So seeing that itself, you can guess what each operation does. So left join, what it does is, it gets all the common values between 2 tables as well as all the values from the left table. So, that is what left join does. And right join, it is very obvious, it will get all the common values between both the tables and also all the values from the right table. If you mention just the inner join, then it gets you back only the common values. It will not give all the values from either the left or right table. It will just give you back the common values between both the tables. If you are performing a full join, it will just concatenate both the tables and give you all the values including the common values. So, that is why the join operation is done.

And as we move forward, there will be so many multi-dimensional queries that we come across in which we will have to join multiple tables in order to extract information. For that, this join operation will be performed.

Next is the ‘group by’ clause. It groups a set of rows into a set of summary rows by values of columns or expression. The group by clause returns 1 row for each group. In other words, it reduces the number of rows in the result set. For example, if see the example that is mentioned below, select status from orders, group by status. So, what we are doing, we are selecting the field which is status from the table which is orders and what we are doing, we are grouping it by status.

So, there will say an order. Order can be having multiple status. For example, if you are placing an order in Amazon, it will initially be in what status? It will be in process status, then it will be shipped, then it will be dispatched. So, there will be multiple status. So, if you are grouping by status, it will give you the output in the form of grouping by the status. That is the orders will be first all the cancelled orders will be there, then there will be disputed orders, then there will be in process orders. So, it will be grouped by the status of the orders.