

Advanced Database Management System_2.1_RDBMS & SQL

Hello and welcome to this topic of RDBMS and SQL. Structured query language, SQL is the language of original database management system. At the end of this topic, you'll be able to understand the various terminologies of Relational model, and you'll also understand features of Relational model. In unit number one, we have discussed about relational model. And we have also discussed the advantages and drawbacks of Relational model. Now it's time to cover the Relational model in more depth. So let us start with the various terminologies of relational models. So in the slide, you know we have come up with a table or relation, and this table consist of the information about various departments. So it's a department number and department name. So it's very understandable that there are three departments which have been shown in the table, and there are few department names.

So the first point is relation, so relation means tables. So in mathematical technology, we say it is a table, but in our RDBMS, it is known as relation, so this is a relation known as department. It's consisting of columns and rows. The second is attribute, and attribute means columns. So, as you can see, there are two attributes. The first attribute, his department number and the second attribute is department name. The third is domain that is set of allowed values. So what values are allowed for the attributes? That is domain. For example, In department number, the values indigent number are allowed and the range we can also specify, if we'll have a attribute as Marks, so we can have 0- 100 that the set of allowed values.

Now tuple are known as the rows of relation. So, for example, this is one of the tuple, 101 MCA or another tuple is 102 MBA. So in former terminology, we also say it is known as records. And rows represent an occurrence is or instances of the table. For an example, I can say, in this department table or department relation, there are three occurrences, 1, 2 and 3. There are three rows or three records. So this table is illustrated to have a clear cut picture and understanding of all that terms, which has been stated in the previous slide. So this is a traditional technologic table in the DBMS, we used to say in relation, but there are different data base manuals, which write the same terminology tables. The columns are known as attributes, and it's also known as field in different data base manuals. Row is known as tuple, and it is also mentioned as records. So, all three are symbols.

Now for the remaining terminology, I have shown these two tables. The first table is department, as you can understand, it consists of three records and the second table is employee, which consists of again three records. So let us discuss the first word Schema. Schema means designing of the table, designing of the data bases. So this database consists of two relations, the First Relation is Department number. First relation is department and second is employee. Now schema means of what kind of designing you are going to deploy. For example, in this department table, we have two attributes, we have specified that domain, their data types, the primary keys, the connectivity, all these are part of designing issues.

Instance means values, values at a particular moment. So for an example at this instance, at this moment of time, this table consist of three records and three values. The second table also consist of three records. So it may change. Instance may change, for an example, if I will have two more employees in this employee table, so the instance will change from 3 to 5. So, instance changes frequently as we apply, insert, delete, update, operations. But schema, once the designing is done, it's going to be fixed. Now degree is one of the very important criteria. It is number of columns. So, for example, in first table the degree is to record. There are two columns and in second table the degree is three. Because there are 1, 2, 3 columns. The Cardinality means number of tuples. So we have in Cardinality equal to three in this first table, and in this table, it is again three. Now relational

databases, collection of normalized relation will have the topics normalization in subsequent lectures. So relation database means if this table is normalized so we can have all the collections is known as relational databases.

Now let us discuss few very important features of relational model. So I have again taken one example of the same department table. So field values are atomic, so it means that if this is a column, so that this is a field values, so this is not divisible, it's going to be atomic, and this is also known as first normal form that will study during normalization. So, let's suppose department name MCA. So we cannot have a classification of MCA into two parts. Alright, it means no common values are allowed. It should be atomic, not divisible.

Now, sequence of column is insignificant. It means either you put the number as first column or on the last column, it doesn't matter. Still, you will have the same answer of your query. Significance of rows is insignificant. It means either the row may consist of first one or last one, doesn't matter, because whatsoever your query will be, if it is satisfying the condition you will have the answer. Each column must have a unique name. You cannot repeat at the same column name with the second one a third one, so it should be unique. Now keys are the backbone of relational databases, so keys it's a fundamental to the concept of relational databases. One of the key that will have in every table is strapping key. It means that it consists of unique values and there is no blank or dull value. As far as this example is concerned, we can take the number as the primary key because it consists of all unique values.

We can also take the name as primary key. But two primary keys are not allowed in a single table. So we have selected the number as the primary key. And it totally depends on the data base designer that he or she is going to select which primary key? Because there may be multiple options. Now the relationship is an association between two or more tables. So that is the beauty of working with RDBMS that we can relate two or more than two tables and we can find out the answer of our query. Now, I have shown this the data may be spread in multiple tables, but we can retrieve the data easily, this is something the best part working with our RDBMS.

For example, we have shown this employee table, employee E1 is working department number three, employee E2 is working in department number five and so on and there is a department table. Department number three is of IT, department number five is of MBA, two is of CSE and four is of MCA. So now the information has been spread, but we can fire a query, from which the answer will have from both tables. For example, if I will say E1 is working in which department? So I want department name. So can we have the letter from first table? So now, E1 is working department number three, you get it. But you need to have a connection from this to this. Then this is known as... given the concept of foreign key joined with the primary table. So E1, once you connect, this E1 is working in IT department. Or otherwise, if I can have a query like this, give me the name of employees; give me the employee number who is working in MC department.

So how you will get it? So, I want employee number. All right, so MCA we have four and what are four? Four is department number. So again, relate over here, and four employees E4 is working. It means that we can connect tables, according to our necessity, and that can be done with RDMS very easily. So this brings us to the conclusion part of this very important topic. In this topic, you have learned that terminologies of relational model and features of relational model, you can also refer the study material which has been given to you. Thank you

Advanced Database Management System_2.2_Relational Query Languages

Hello and welcome. We are now going to discuss the topic of Relational query languages. At the end of this topic, you will be familiar with the QBE, SQL and ISBL platforms of RDBMS languages, and you will also learn the components of SQL. So let us start with QBE. QBE stands for query by example and was developed in 1970 by Moshe Zloof at IBM Research labs. Now, it provides a graphical user interface because with working with SQL you need to understand, you need to know the command structure of SQL. Then only you can retrieve whatever you want to have with your query. But in QBE, you don't require so specific SQL statements knowledge. There may be graphical user interface, and that is very helpful to retrieve the answer of your query. You can also remember Microsoft Windows Operating system. Everything is based on GUI. One of the best features of windows, so it's on the same pattern we can have QBE, working with RDBMS language, so it removes the conditions of writing SQL queries, already discussed.

Now the best thing is that QBE is nowadays, it's supported by most of the database programs, and it makes the easiness for database applications, either for creation or modification or retrievals. Now let us focus on the example. Suppose you want to search employee record, who is working in MCA department, so, it means that there exists a table of employees of different departments. Now, if you want to have the result of these employees, so in a SQL you need to write the query, select employee name, employee ID. Then from table name, and you need to put the where clause, if it is required. But in case of working with QBE the only thing is that you will be provided with the proper interface, and you need to select what is required, and you will have the results very soon. So QBE is very, very useful as far as easiness is concerned with working with RDBMS languages.

Now let us explore SQL, so SQL stands for structured query language, which is the standard database language used to create, maintain and retrieve the data from the relational databases. It has been widely used. So, there are different vendors, who have implemented the SQL platform like MySQL, Oracle, Microsoft SQL Server, Host base, Sybase, Extra. Now SQL is not case sensitive. It means either you were right in lower case or upper case, so the commands will execute without any errors. But in various databases, manuals and books, you'll find that the SQL commands are written upper case. So, it just for the sake of clarity.

SQL programming can handle large records and manage numerous transactions. So suppose if your database consists of 50, 60, 70 tables, so there is no issue with that. You can manage that databases and transactions data, that is because transaction data may have insert, delete, update kind of statement that can be very easily manageable with a SQL. Now it supports data independence feature. So as a user, if you going to fire any query, you are not bothered about from where you are getting the results of the query. So that is data independence. It means that you are not familiar about the other details, the schema details, the relationship details, the physical details, where the data has been stored, where the table has been stored. The only thing is that you have been provided an external view. Where do you put your query and you will get the response.

Now SQL has different components. The first is DDL, which stands for data definition language. It is used to define the database structure or schema. If you want to create the database, if you want to create tables or if you want to create views, then you will use this data definition language or specifically create command. So this few examples of DDL commands are create, alter, drop, truncate. We will discuss each of these commands in the subsequent sessions. The next is DML, which stands for data manipulation language. It is used for inserting or deleting or updating data or tuples in the databases, and the few commands are insert, update and delete. The next component of SQL is DCL, which stands for data control language. It is used to control the excess of data stored

in a database. Suppose I am working as a DBA, so I may control the other users, which privileges provides to other users based on their requirement.

So there are two commands grant and revoke of DCL. The fourth component is TCL, which stand for transaction control language. It is more specific towards data programmers and DBA. It is used to manage transactions in the database, and examples are commit, roll back and save point. The next and last is DQL which stands for data query language, it is used to retrieve the answer of your query. So if you need to write any query, you need to use this select statement and then based on the Syntax, based on the data availability, you will have the answer of your query. So this is just a snapshot of the components of SQL commands, so we'll see more on SQL commands in detail with examples in the subsequent sections.

The next language. I wish to take care of is ISBL, which stand for information system based language. It's a pure, rational, algebra based query language, which was developed in IBM Research Lab, UK in 1973. So we have two mathematical models, for implementing SQL Languages. First is relational algebra and second is relational calculus. So ISBL is purely based on the relational algebra. So, it was first to used in an experimental interactive database management system called Peterlee Relational test vehicle. In short, it is PRTV using ISBL. A data base system can be created with the size of about 50 relations. So you can have Max of 250 tables, and each table may have maximum to 65,000 tuples. So this is one of the limitations of ISBL. The relation in ISBL can have utmost 128 columns. So these are a few properties of ISBL. So this comes to the conclusion of this topic. In this topic you have learned about types of relational languages. You have learned about features of QBE, SQL and ISBL. You can refer the study material provided for more information. Thank you.

Advanced Database Management System_2.3_SQL

Hello and welcome. Today we're going to discuss the topic of structured query Languages, SQL. So SQL is a very standard language of RDBMS used for database creation, any modification required and the retrieval of data from the entire databases. So at the end of this unit, you will be able to understand the additional features of SQL. So let us start the session. As already, we have discussed in the previous sections, about SQL that it's a standard data base language for creation, maintenance and retrieval of databases. There are different vendors like MySQL, Oracle, Microsoft SQL Server, and Sybase. It is not case sensitive. It can handle various transactions, optimization, it can handle so much large number of tuples or records also, and it's supposed data independence features.

So every point has been explained in the previous section. We have also discussed the components that sql have majorly five components, DML, DDL, DCL, TCL and DQL. So we have discussed few examples in the previous session and in the upcoming sections also will discuss the different queries with certain examples. Now let us see our various additional features that you need to understand one by one. The first is DML, DML stands for data manipulation language, and it is used to manipulate the existing tables or databases. So if there's a table exist, so all the way you can insert the records, you can delete or update new records.

So we have the proper Syntax for that, proper command structure for that, and based on that, you can fulfil the requirements. The second is data definition language. It is used to create schema or database or tables. You need to use the create command, and also we have alter command for any changes and drop command if you want to drop or read the table, and we also have the truncate command. So we'll see the differences in terms of SQL syntax, in the subsequent slides. Now, in sql we have the facilities of triggers and integrity, constraints also. So triggers that whenever some event happen, so DMS will fire a message. So as one of the example I can give you that, whenever

you withdraw something from your ATM, so immediately there's a message so that kind of coding can be part of SQL.

So triggers are the very useful applications working with databases, and you can have the formal definition. Also, that triggers are the actions that are run by DBMS, whenever some event related to database occurs. Now SQL also support the run Time or dynamic and embedded SQL. It means if you want to work, retrieve the SQL statements that are part of some other host languages, like C or COBOL or C++, then definitely, you can execute the runtime query and you will have the results and the last year's execution of the client server application and remote access databases. So this is something a very useful feature, where a client program can connect with the server database and allow the user access for the more database access.

Now transactions, they are the heart and soul of DBMS applications. Transactions have certain properties of atomicity, consistency, isolation and durability. So all these properties, apart from the start transaction and end transition that can be variable supported through SQL commands, sql also offered high level of security and to privilege access. So it controlled the access to tables and used, thereby protecting the databases. And many features such as recursive and decision support queries, object 200 features are provided by sql. As we know that we have, O RDBMS, that is object relational database management system, an object oriented database manual system, so in which we have acquired the concept of classes, objects, inheritance. And that makes sense because working with a sql you can have; all the way you can have additional features. So this comes to the end of this topic. We have discussed some additional and valuable and rich features of sql. You can always refer the study material, which has been given to you, for getting the in depth study. Thank you.

Advanced Database Management System_2.4_Integrity Constraints

Hello and welcome. Today, we're going to discuss the topic of Integrity Constraints. So in every relational database management system, you may find that integrity constraints exist, and they maintain that there should be no wrong information, shall be a part of any of those relations. So the data integrity, they are the very essential components of working with RDBMS. So at the end of this unit, you will be able to understand integrity constraint and their types.

And so let us start the session. So we may have three types of integrity constraints. So let me make you very clear about the word constraint. So constraint means conditions. So what are the conditions were are going to impose in that table, once we're going to define it. All right, so the first one is entity integrity. And as far as entity integrity rule is concerned, it is applicable in the concept of primary key and primary key is a column which may must be unique. It may have unique values, and it cannot be blank. So this is known as entity integrity rule. I'll repeat primary key must be unique, and I will show the examples right now and the second is family key cannot be null value.

So I have referred a table from customer details record, from customer details table. You can see there are different attributes. Customer ID and customer last name, mid name, first name and their account number. So this has been highlighted, because it is taken as primary key. All values are unique and we have other columns as account type and the branch and customer email. So you may have opportunity to select other columns as primary key also, for example, this customer id is also a candidate for primary key. But in a single table, we select only one Primary key. If you have more candidates to become primary key, it is known as candidate key, all right? So in this table, account number has been taken as the primary key.

Now, as far as entity integrity rule is concerned, the primary key must be a unique. So you can say all values are unique and second there is no null value. So, null itself is a very valuable concept in a SQL terminology, because null is not equal to zero. Zero is different, and null is different and null maybe you know the blank values or missing values, that we can have in future. So it should not be like that. So primary key should not take repeated value, it means it must be unique, number one. And second is the primary key should not take null value. So this is known as entity integrity rule. You can also take example of some real world concept, that if I'm going to maintain the records of 60 students, so if I'm going to have a table for that, so I will make their enrolment number as a primary key. So as soon I will make the enrolment number as the primary key. So every student must have that enrolment number, and second is all this will be unique, and there should be no blank entries in that column.

So this happens, in practical scenario, like some of the examples are aadhar card numbers. And in railway reservation we have PNR numbers, which is always unique. So there are lots of examples. So as a database programmer or DBA, you need to have a thought on that, which is the more suitable column to become primary key. The second integrity constraint is known as domain integrity, which is also known as attribute integrity. It's a group of allowed values for any specific column and the values maybe size, you know, and data types and null values, all that. And how we implement this, it can be with default constraint. Alright! And you can implement this with foreign key or check constraint and data types. So that is domain integrity. Let me go through one example.

So I have these two tables, the first table is department table and second is the employee table. Right! So as far as like, suppose I want only integer value over here, right in this attribute so it is there, that we can impose. Or supposed in this, in the D number-department number in this employee table, null values are allowed, so that is domain integrity. So we need to define it, and we need to be very clear about the requirements and the working procedures. So that's why we will be able to define it. So that is attribute integrity. It means what values are permissible and what values are not permissible. The next is referential integrity rule. It means when a table refers to other table, then we apply this referential integrity rule, and it says that foreign key, I will let you know what the meaning of foreign key just now is. So foreign key must either be null or match the values in the relational table, in which that foreign case linked to the key column. So let me go through this example, it will make the concept very clear.

So this is our first table- department, this is the second table. And you can see there's a column department number over here and there's a column department number over here. And what is the meaning of this? The meaning is department, there are three departments and their names are given. For example, when 101 belong to MCA, and the second table Employee, we have three. You know it's all about employees, employee, their name and which department they're working. For example, like if you take this example, employee number three, so employee number three is working in department number 102. So 102 link to this column, parent table column and 102 is MBA. So if I can have a question, that employee number three working in which department? So you can see it's working in MBA department. So this is known as for foreign key concept.

So this column, that D number is known as foreign key, and it refers to this primary key of this parent table. I hope you got the idea. So this referential integrity says that foreign key values can be null. For example, this, like employment number one, his name is A, but he is not working any other departments. So its null is allowed in foreign key. And second is foreign Key should have the value, which refers to the parent one. It means foreign key cannot have some different values, which are not presenting the parent column, like suppose if I can have 101, it's okay because it is in the parent. 102 it's okay, but I cannot have 104, 105 because they're not in the parent column. So there are two

rules. I repeat. 1st is null values are allowed, and second is foreign key values shall be the values, which is from parent table. Not beyond that. So this is known as referential integrity rules.

Now, as for the sake of getting it more clarity, I have taken one more example from a bank database. So the first one is a table account number, branch name and balance. And the second is branch, which has the information of branch name, building, street and city. So you can understand that how this has been made, us foreign key, because branch name, this is the table refers to. You can see the arrow, refers to the branch named after some other table. All right, so this branch name is foreign key, because it refers to this parent table. All right. Now, this referential integrity says that, there should be, null values are allowed, so we don't have any null values, doesn't matter. Duplicates are also allowed in foreign key, so they're duplicates, they're allowed.

And third concept is this branch name should be a part of this, we cannot have any branch name which doesn't exist in the parent column. So let's see. The first one is Kanr Main, it's there. Second is CLT main. It's there. Alright, CLT main, it's there. And third is Clt sub, it's there. So all the values in this foreign key column are a subject or equal from this one, the parent column. So this satisfied the concept of referential integrity. Let us check the first one. It means that the integrity constraint, integrity constraint. The first is all values are unique, so they have unique values and there is no null value. So I think based on this example, you've got the answer of integrity constraint.

So this comes to the conclusion of this topic. So in this topic, you learned that was the need of integrity, constraint in databases. And we have also seen the three types of integrity constraints such as entity, domain and referential integrity rules. So this comes to the conclusion and you can refer the study of material provided to you for more information. Thank you

Advanced Database Management System_2.5_DDL

Hello and welcome. Today we're going to discuss the topic of Data definition statements. So DDL is a part of SQL which is used to define the database structure, or schema. At the end of this topic, you'll be able to create relational database relations using SQL, and you will be familiar with the Create Alter Drop Truncate commands of Data definition language. In the previous sections, we have discussed SQL components, so we have discussed about DDL, DML, and DCL. We have also discussed TCL that is transaction control language, and Data query language. So already all this has been discussed in previous sections. Now let us explore every component on subsequent sections. So let us start with data definition statement. It means the DTL commands. So we understand that create command is to use create tables, not only tables. It can be used to create databases, tables and schema and views.

So, it's very analogous to the English words. SQL is so analogous. Because you understand about meaning of the create words. So same create command can be used for creation of tables also. Now we will see the example and exactly how it works in the subsequent slides. Now Alter command. It's used to modify the structure of the database. The drop is to delete the tables from the databases and truncate is again to delete the information from the tables but there exists a difference between drop and truncate, that we will discuss soon now. Now, let us start with the first command that is, create table command. So create table command. If you want to create a table so we can start working with this and the Syntax of the create table, and already you know that it's not case sensitive.

So either we can write in upper case or lower case or the mixed version of both. It doesn't matter. You will have the proper execution of the command. So create space table, and then you need to write the table name and bracket open, and in that part you need to write column 1. Which column

you want and what is the data type? What is the size and constraint, if it is null or not null, and then you can repeat as many columns you have. Let me show you the example. So this is a create table command. It is used for creation for employee table. So based on this, the schema of the employee table will be created. Now all this employee number is a column name or attribute name. Number 4 is a data type, we have used with SQL and NOT NULL. So what's the meaning of this? It means there will be a column known as employee number, and it will accept the number values up to four digits, and it will not take blank values.

Alright, so that's why it's NOT NULL. Second, we have taken employ name. So this is as VARCHAR 2. This is one of the Data types which accept characters. It is of length 10. We have taken job as again VARCHAR2. And the length is 9. And the date of joining. We have taken date, so Date is one of the data type available in all SQL's. SAL means salary. It is a number that you know 7 commas decimal 2. Then commission number and department number. All right, so we have taken like this also. So as soon we'll execute this with a SQL platform. It will create the employee table with all these columns. So table schema will be done and then we need to insert the values which are required for these tables. Next is Alter table command. It is used to add, delete or modify the columns in an existing table. The syntax is very simple. You need to use the keyword alter, space table and then table name and what you want to do.

Suppose I want to add this column, so I will just right add this column. So let me go through one example. So Alter table employee means employee table already exists. And I want to add one more column, salary. So add salary. And what is the data type of this. It's integer 5. As soon as you will execute, so the existing table employee will be have on additional table known as salary. Let us move through the next command. That is truncate command. Truncate command is used to delete the complete data from the existing table. And if you want to restore, because as soon as you will just apply Truncate, so all rows, all Tuples all records will be deleted, but the schema will still exist. Schema will not be deleted. The design will not be deleted. And if you want to have the undo, if you want to have the information, all information again, you can use the rollback command that is from TCL transaction control language and it will restore the entire tuples.

So in truncate information can be deleted, but schema exists. Schema persists. So this is the format, Truncate, then space table and table name and as far as example is concerned, suppose if I write truncate table department, so the department table, all information will be deleted. Otherwise, I can say all records of this department table will be deleted. The same purpose can be achieved through the Drop command. So, Drop table command is used to delete all the tuples as we have seen in the previous command Truncate but Drop Table deletes everything. The data as well as schema. Nothing is left out. So that's the second point, drop the whole data and schema of the relation. And here we cannot restore by using rollback Command, what we have seen in the previous slide or the truncate command. So there's a format; Drop space, table and table name.

And there is this example. As soon you will just use this drop table employee so the entire employee table will be gone away with its structure and you cannot recover it. Now we have two types of drop command. The first is Cascade. So it deletes the complete database schema. Whatsoever elements exist in the schema, it will be deleted. And if you use the restrict command, so it will just delete the database schema. If it will not have that element, it will not be terminated. So we have these two options, Cascade and restrict. So this brings us to the conclusion of this very important topic of DDL. I hope you have learned the various commands and with the examples. So we have discussed about scope of Data definition language and they have also seen how to create new tables, how to alter existing tables and we have seen delete existing tables. You can also refer the study material provided to you for more information. Thank you.

Advanced Database Management System_2.6 DML

Hello and welcome. Today we're going to discuss the topic of data manipulation language. So the DML, in short, is used for any changes that you need to undergo with SQL statements into the database schema. So at the end of this topic, you'll be able to have the familiarity with the DML commands. You'll understand simple and sub queries. So let us begin the session with the commands of data manipulation language, so the first command is insert. Whenever you want to add a new record in an existing table, then you can use the insert command. We'll explore different examples in the next slide of insert command. The next is delete; when you want to delete the records from an existing table, then you can use this Delete key word. If you want to update a field value or change values, then you can use the Update Command.

We'll also explore the Select Command, which is also known as data retrieval or data query language, and it is used to fetch the data from one or more tables. So let us explore all these commands with their features and examples and their Syntax. So if you want to insert records in existing table, then we'll use the insert command. And Syntax is very simple. You have to use the insert. Again I repeat that SQL is not case sensitive. Just for the sake of getting it in an easy manner, I have written everything into capital letters. But if you type in smaller letters, also, it will work with sql editor. So the syntax is insert into. Here, you need to specify the table name, then what values you need to enter in all the columns.

Let's see the example. And so insert into employee that table name and values so it will take values according to the columns. So it means that first employee number, then name and then position and so on. So all columns will have the values as shown over here, but there may be number of cases when you don't want to insert all values or values in all columns, then you can try it like this. So insert into the employee table and then you write just a column name, where you want to insert the records and then with the values. Plus you need to show the values. So what will happen, so the values will go to this column? It means employee number will have 101, department number will have 29 and ename will be Sujit.

The next command under DML is, if you want to delete any existing records from a table. For example, if you're dealing with the employee table and there is an employee who has left organization, now you have to delete the records or the data of that employee. So you have to use that delete command. So delete is used to delete records from existing table and the syntax is this, delete from, here you need to specify table name and where a clause and the condition are. So this is an example, delete from employee, where employee ID equal to two. What will happen, as soon as you will execute this command so employee ID two will be deleted?

Now you can also run this command delete from table name, so it will delete all rows from the table. So if suppose, for example, if you write delete from EMP, so all rows from employee table will be deleted. Let us move to the next command. If you want to update any field value, then you have to use the update command. So that's why the update statement is used to modify the existing records in a table and see this is the Syntax of the update command. Update space and followed by table name. And then you need to specify the set column one, what value you need to enter, column two what's the new value and where is the condition? Let's see the example. Suppose I need to update the employee table and suppose employee ID four has salary 4000, now the salary we need to increase and we need to set the new salary of 6000.

So I have written update Employees set salary equal to 6000. Where employee ID is equal to four. As you will execute so immediately, employee id four, salary will be changed to 6000.

Now let us go ahead with the very important Syntax of select statement. So Select is used to retrieve or to fetch data from existing table or more than one table. It's a very useful and if you have professional skills working with SQL, so you must have that kind of expertise with select statements. So let us go ahead with the fundamentals of select statement one by one. It is used to select data in rows and columns from one or more tables. Now every statement in this is known as clause. So there's the first clause Select. Now you need to specify what are the, columns you want in the answer and from table name? So it means that in that select statement, if you have minimum these two clauses, first clause is this and second is this. If you have two clauses, you will have the answer.

Let us see the example so asterisk is used for all. Suppose you are not giving the column names, so if you type a strict so you will have all the columns from the existing table and from table name. Now there's a third Syntax of select Command, so select against same first clause from now. If you want to impose certain condition row wise, then you need to use- where. Group by if you want to group according to certain conditions and order by is for, like you want the answer in some order, either ascending order or descending order. So these are the Syntax we can refer and to see some of the example that will make the concept very clear to all of you.

We can also use various operators embedded with SQL statements. So these are various mathematical operators that can be used as a part of SQL query. I hope you understand the meaning of all. Either it is greater than or greater than equal, less, less than equal or not equal to. It can be a part of sql query, as well also supports different logical operators like and or not and means both conditions should be true or means anyone of that should be true and not means it's not going to be true, and it also supports various arithmetic expression like plus, minus, asterisk and divide. So once we're going to write SQL queries using select statements, we can use any of that operator, which has been shown in the slide.

Now let us go ahead and just try to understand that various examples. So the first example is an employee table. So as you can see, there is an employee table, which consists of total six columns, so columns are also known as attributes. So we have the six attributes and name of table is known as relations. So this relation name is employee. Now how many records we have? 1, 2, 3, 4 we have four records, and this table is all about employee details, and it's showing their designation, their department, their salary and their incentives. Now, if we fire inquiry, select employee number, ename and department number. So you understand whatsoever column we have shown that will be a part of answer. So we have required these three columns that what is the employee number, what is their name and what is Department number.

Now select from which table you are referring to. The table is employee and where Department number equal to two. So this is a row wise condition that we want the record of only those employees who are working with Department number two. In many cases, like suppose in any university, suppose you want all the records of CS Department or IT department. Then you can simply that department name equal to IT or department in equal to MCA. So you will have all the records of employees who are working in that department.

So where clause is very important and you can put the conditions in their clause and you will have the answer. So as soon this query will be fired, you will have this table as the answer, on three columns will be their employee number, employee name, department number, and you will have the records. So Department number two, you have only one record. And this has been in the answer. But you can see that there are total six fields, or six columns, but they are not in the answer. So it will be only three, which has been mentioned in the select clause. If I have written asterisk over here, then we will have all six columns in the answer, with this one record, right?

Let us go ahead with the second example. Now here we are writing select. So, asterisk means all and from employee tables their designation equal to manager. So there are all you know, manager. So we have mentioned this, that the table consists of all that because asterisk is there. So we have all that you know columns and whoever is satisfied, the term manager will be part of the answer. So by this way sql select statement works, it's very important if you if you have created schema, if you have inserted the values, then the next very important aspect is that how to retrieve data.

So to retrieve data, you must understand the proper Syntax offset command. Then only you'll be able to get the data from existing table. Now let us go ahead with the concept of subquery. These so all the way might be in C and C ++. We work with nesting, nesting of function. So same concept has been applied over here and that is known subquery. It is select statement embedded in another select statement. So we have inner query, that is known as subquery and it executes before the main query and the result of the subquery is used by the main query and you can all the way use single row operators for single sub row queries.

I will let you know all this concept through the example. Now this is example. First of all, let us clear with the fundamental terms, which we have discussed earlier. So this is a select statement inside the select statement, so this is known as inner query. Alright, so whatever the result of the inner query will be, it will be given to the outer query. So this is the concept of subquery. In different interviews, I have seen that they asked for, find out the second higher salary. Find out the third higher salary and you need to use subquery. This is very easy, once if you are select statement, and based on the logic you can find out or you can satisfy all that conditions. So see the query is which employees whose salary is better than employee 'A' salary? So, first of all, you need to find out what the salary of the person employee. So this is the inner query. Select salary from employee, now where is used for condition, so employee name.

Now, once you suppose the answer of this inner query is 20,000. Now you want all those employees whose salary is higher than 20,000. So that's why the Syntax, higher or greater, and then you can write, select employee number or that all details you want in employee from employee table, where salaries higher than suppose you got result 20,000. So this is one of the concept of subquery, they are very, very useful for going through some sort of complex queries, and by just working with software is we can get the results in a perfect manner. Apart from that, we have some group functions and let us discuss the first group functions as aggregate functions.

So basic informed, we have total five aggregate functions, count, sum, average min, max and the meaning exists as the word is. So count means that it will count the number of records. Sum is the summation part, average and minimum is whatsoever the minimum value exist and the maximum, what is the maximum value? Apart from this etiquette function, we have the group by function also. So as you expand the select query or the select statement, we have already seen select from where now you can also use group by, and having. So having is a condition which is imposed in group by and then you can use order by.

So don't worry, we have examples for all these queries. Let us go through one by one. So Count is used to count the number of tuples in that column. It's very clear. Let us see this example. Select, count, asterisk, from employees. So if you'll just queue the counting, how many tuple exist in this employee table? Now sum is written as with a column name, and it gives us sum of all tuples present in that column. Average is used to find out the average of a given column. We will explore the example soon in the subsequent slides. Maximum returns, the maximum value inside a column and minimum returns the minimum value inside a column. So let us see the examples of all five. So I have taken one relation and relation name is employee. Let us get familiarity with this relation.

There is employing ID. There are four employees, employees name A, F, R, D, and there some salary which has been shown in the table. Let us put some queries. So the first query is, select count* from employee so * means all also, you need to count how many tuples exist in this, so it will just give the counting and you will have the result as four. So Count* it is four.

Let's see the next query. Select some salary, so you understand the syntax also. You need to write aggregate function and the bracket you need to note down the column name or attribute names, select some of salary from employees. So what is the sum of all these things? It's 10,000, so you will get the answer as some salary 10,000. This output you will get from masculine. The next is select every serving from employee, and it's very simple, if you just find out the sum divide by four. So the average salary is 2500. I hope you understand the concept of Aggregate function.

Let us go through the remaining two functions. So I have repeated the table. Let us see the query; you need to find out the maximum, so select max of the salary from employees. And the result is this, that the maxim salary is 4000. Similarly, we can go ahead with the minimum salary, to select min salary from employee and then we have the answer. So how the aggregate function has been used, it is used with the select clause in the bracket, the column name and from which table you want to refer. Let us also see the group by function, which is very important.

So we have used a select statement again. Select First Column. I want this department and then sum of salary. So it means that I want to group all employees according to their department, and then find out the salary of the entire department. So select, then from employees, from for what you're going to a group, so by departments, the all that employees who work in the same department will be grouped, and you will show in the results. Let us explore the results. So the result will be like this. First will be department so and then salary. And this salary is a summation. So for an example for the electrical department, like they're May possible, that there are five or six people working, but they're sum of the salary is 25,000. Similarly, the salary of the sum of the salary of electronics department has 55,000 and sum in aeronautics is 35,000 and some in the department InfoTech is 30,000.

With this, we come to the end of this topic that is data manipulation language. In this topic, you learned about DML features and various commands like insert, delete, update with examples. We are also explored Select statement, aggregate functions and we have seen concept of subqueries. You can refer the study material provided to you for more information.

Advanced Database Management System_2.7 Views

Hello and welcome. Now we are going to discuss the topic of views. A view is actually a composition of a table in the form of predefined SQL Query. So at the end of this topic, you will be able to understand the concept of view and you'll be able to create views using SQL. So let us explore some features of view, view stand for virtual table that can be created from one or more than one tables. Views, obviously which are type of virtual tables, are used for multiple purpose. Let us explore. First is it can structure data in such a way that user feels that is of more easiness because working directly with table may have its own pros and cons.

So instead of working with the entire table, if I need some, certain part of that table so I can create views and that can be source of my working. So its structured data in a way that users or classes of users find natural and easy. The second point is it restrict access to the data in such a way that user can see and modify exactly what they need, and know more. So it's again same that if there are, say, 12 columns in a table and you don't want to work with each and every column you want a

fragmentation of that table, so you can use a views according to your requirement. And third is it is used to summarize data from various tables, which can be used to generate reports and so on.

So let's see how to create the views using sql. So all of us know that the create command is used for create of tables. The same SQL Command create, is used to create views. So this is the Syntax. Syntax is like this, you need to use create, then space that view, you need to write. And then you need to specify what is the name of the view. So view name as and then you need write SQL query. So select what kind of column you want in the view. What is the source that is from table name, and then where is what's the condition? So you need to be very clear about this SQL statement and whatsoever the results will be, it will be pre store in terms of this view.

Let us see one example. So I have created a view, and this is the entire name is of that view. So my view name is department IT 101 employees. Now how come I'm going to get this view? So I have to written A as select* from employees, their department ID equal to 101. So in the employee tables, wherever department ID 101 all that information will be stored in this view, which is this department ID 101 employees.

So suppose I need to explore the different details, I need to add something with his employees, so it will be very easy that I'm working with a virtual table and working with the subset of the main table, and I can directly explore the concept of views. Let us see if you have created the view how to use it. So it's very simple. It's just analogist to using the SQL statement. So select asterisk from the view name, so you will have all the results of the view that have been stored earlier. Now DML operations on the view is DML operations on the table because already I have communicated that view is a subpart of the main table. So if you're doing some DML operations like insert, delete, update so that will be also reflect in the main table.

Let us see this example. Suppose I have inserted this insert in to this view which I have just created in all the columns which I wish, and I have entered this value. So do understand that this will be part of the view and at the same this data 25 Achintya, 10,000 and 101 will also be stored in the main table, so we can have this result one row inserted into the employee table. Because it's not that you're working only with this view. It's attached directly with the employee table, so you will have one row inserted into the employee table.

We can also use deleting few records from the view, very similar to the same concept of deleting few records from the table. So we can write, delete, from this is the employee view. I have already created where employee ID are equal to three, so wherever it satisfy row wise, this you know that record will be deleted. We can also use dropping a view, where all that information of the view will be gone away, it will be deleted. So if you write this, drop view and then your name and example drop view, employee view, so all that view will be gone and there will be no existence of the view which we have created. So this is all about deleting the view and dropping everything.

So this comes to the end of our topic of virtual table. In this topic, you learned about the concept of virtual table. We have also seen how to create views, how to use a view, how to delete records from a view. And we have seen how to drop the entire view. You can always refer the study material provided to you for more information.

Thank you

Advanced Database Management System_2.8_Embedding SQL Statements

Hello and welcome. We are going to cover the topic of embedding SQL statements. So embedded SQL statements are those SQL statements written in line, with the program source code of the host

language. At the end of this topic, you will be able to discuss the concept of embedded SQL, and you will also learn syntax of embedded SQL. So let's start with the features of embedded SQL statements. So you have seen different SQL statements. Now let us see how it can be clubbed with other programming languages. So this is one of the features that sql statements can be embedded into various types of programming languages.

Now what are the programming languages? And they are known as host language. So the next point is host language is the language in which the SQL queries are embedded and I have listed certain examples like C, C++, FORTRAN and pascal. They are the very old languages, not currently in very much used, but still we had that feature that they can call sql statements and they can embed sql statements. The sql structure, which is embedded in the host language, is done as embedded SQL. And soon we will see the Syntax how it's going to be happened.

And this is a syntax, that EXEC space SQL. And we need to write the embedded SQL statements and END-EXEC. Now let us see one example that will make the things very clear. Let us consider some banking schema and we are working with the host language like C or C++ and we have defined a variable that is amount, all right. Now we need to find out the customer records or the details of customers who satisfy certain conditions.

I will let you know with the example what are the conditions? Now for this we have defined a structure like this. So we have defined a cursor and name of the cursor is C. So there's a Syntax declare c cursor for. Now, we have written the SQL statement that we require customer name and customer city from these two tables, deposit and customer. Where deposit thought customer name equal to this, customer dot customer name and reported Balance is higher than amount. So what is this? So, there have been certain criteria and amount is a variable and we need to find out those customers who balance higher than the given amount.

So the query evaluation is like this **(unclear)** SQL, that open c or d, already in the previous slide, you have seen how cursor has been defined. We can also fetch the cursor in part of embedded SQL. You can see this fetch c, already defined in the previous slide into this there are two variables, customer name that is cn and cc. Once you will have this value, now you can change and alter according to the host language Syntax and then you can end. You can also close a statement with the same Syntax what have used for Open. So it is EXEC SQL, close c END EXECUTION. And you can also modify the records with the statement, what we have mentioned with the next one and very **(unclear)** statements or whatever Modifications So the s e C s will and valid no update inseparably come in and you can close with and the SEC lakhs index we use when we call different emigrant you know ask your statements in different most languages.

So this brings us to the completion of this topic of embedded SQL statements. So in this topic you have learned about embedded SQL, you have also seen the syntax of sql. For more information, you can refer the study material provided. Thank you.