INTRO. TO DBS

Lecture 4

Data is everywhere lets manage it!

SQL

- Structure Query Language (SQL)
- Programming language storing and managing data in RDBMS.
- Ist commercial language introduced for E.F Codd's Relational model.
- Almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) uses SQL as the standard database language.

DDL : Data Definition Language

Command	Description
create	to create new table or database
alter	for alteration
truncate	delete data from table
drop	to drop a table
rename	to rename a table

DML : Data Manipulation Language

Command	Description
insert	to insert a new row
update	to update existing row
delete	to delete a row
merge	merging two rows or two tables

TCL : Transaction Control Language

Command	Description
commit	to permanently save
rollback	to undo change
savepoint	to save temporarily

DCL : Data Control Language

Command	Description
grant	grant permission of right
revoke	take back permission.

DQL : Data Query Language

Command	Description
? select	retrieve records from one or more table

alter command

- Alteration of table structures
- to add a column to existing table
- to rename any existing column
- to change data type of any column or to modify its size.

alter command

- adding a column to an existing table
  - **alter** table *table-name* add(**column-name** *datatype*);
  - alter table Student add(address char);

- The above command will add a new column *address* to the **Student** table

alter command

- adding multiple columns to an existing table
  - **alter** table *table-name* add(**column-name1** *datatype1*, **column-name2** *datatype2*, **column-name3** *datatype3*)
  - alter table Student add(father-name varchar(60), mother-name varchar(60), dob date);

- The above command will add three new columns to the **Student** table

alter command

- adding a new column to an existing table with default values
  - **alter** table *table-name* add(**column-name1** *datatype1* **default** *data*);
  - alter table Student add(dob date default '1-Jan-99');

- The above command will add a new column DOB with some default value to the **Student** table

alter command

- modifying data type of an existing column
  - **alter** table *table-name* modify(**column-name** *datatype*);
  - alter table Student modify(address varchar(30));

- The above command will modify *address* column of the **Student table**

alter command

- Using alter command you can rename an existing column
  - **alter** table *table-name* **rename** old-column-name to column-name;
  - alter table Student rename address to Location;
- The above command will rename *address* column to *Location.*

alter command

- dropping columns
  - **alter** table *table-name* drop(column-name);
  - alter table Student drop(address);
- The above command will drop *address* column from the **Student table**

truncate command

- removes all records from a table. Without destroying the table's structure.
- When we apply truncate command on a table its Primary key is initialized.
  - **truncate** table *table-name*
  - truncate table Student;
- The above query will delete all the records of **Student** table.
- Different from **delete** command. ??

truncate command

- Different from **delete** command. ??
- **A** table with 10 rows with an auto_increment primary key, with *delete* command to delete all the rows, it will delete all the rows, but will not initialize the primary key, hence if you will insert any row after using delete command, the auto_increment primary key will start from 11.
- But in case of *truncate* command, primary key is re-initialized.

drop command

- ??
  - **drop** table *table-name*
  - drop table Student;
- The above query will delete the **Student** table completely.
- drop database Test;

rename command

- Renaming a table
  - **rename** table *old-table-name* to *new-table-name*
  - rename table Student to Student-record;
- The above query will rename **Student** table to **Student-record.**

DML command (**INSERT command**)

- inserting data into a table
- **INSERT** into *table-name* values(data1,data2,..)
- Consider a table **Student** with following fields.

S_id	S_Name	age

INSERT command

- INSERT into Student values(101,'Adeel',15);
- The above command will insert a record into **Student** table.

S_id	S_Name	age
101	Adeel	15

UPDATE command

- Updating a row of a table.
  - **UPDATE** *table-name* set column-name = value *where* **condition**;Lets see an example,
- ?? ; For the given table Student??
- update Student set age=18 where s_id=102;

S_id	S_Name	age	S_id
101	Adeel	15	101
102	Aslam	18	102
103	Kamran	14	103

Update multiple columns

- UPDATE Student set s_name='Amjad',age=17 where s_id=103;
- The above command will update two columns of a record.

S_id	S_Name	age	S_id
101	Adeel	15	101
102	Aamir	18	102
103	Amjad	17	103

Delete command

- Delete data from a table
- Delete command can also be used with condition to delete a particular row
- **DELETE** from *table-name;*

Delete command

- **Delete a particular Record from a Table**
- Consider the following **Student** table

S_id	S_Name	age
101	Adeel	15
102	Aamir	18
103	Amjad	17

Delete command

- DELETE from Student where s_id=103;
- The above command will delete the record where s_id is 103 from **Student** table.

S_id	S_Name	age
101	Adeel	15
102	Aamir	18

WHERE clause

- To specify condition while retrieving data from table.
- Mostly used with *Select*, *Update* and *Delete* query.
- If condition specified by ***where* clause** is true then only the result from table is returned.

WHERE clause

- *SELECT* column-name1, column-name2, column-name3, column-nameN from table-name **WHERE [condition]**;

WHERE clause

- Consider a **Student** table,

s_id	s_Name	age	address
101	Adeel	15	Pindi
102	Aamir	18	Wah
103	Amjad	17	Attock
104	Aslam	22	Hassan

WHERE clause

- SELECT query using WHERE clause.
- For the following table Student.
- SELECT s_id, s_name, age, address from Student **WHERE** s_id=101;

s_id	s_Name	age	address
101	Adeel	15	Pindi

SELECT Query

- Select query is used to retrieve data from a table.
- We can retrieve complete table, or partial by mentioning conditions using WHERE clause.
- **SELECT** column-name1, column-name2, column-name3, column-nameN from *table-name*;

SELECT Query

- Conside the following **Student** table,

S_id	S_Name	age	address
101	Adeel	15	Pindi
102	Aamir	18	Wah
103	Amjad	17	Attock
104	Aslam	22	Hassan

SELECT Query

- SELECT * from student; The above query will show all the records of Student table, that means it will show complete Student table as result.

S_id	S_Name	age	address
101	Adeel	15	Pindi
102	Aamir	18	Wah
103	Amjad	17	Attock
104	Aslam	22	Hassan

SELECT Query

- SELECT s_id, s_name, age from Student.
- The above query will fetch information of s_id, s_name and age column from Student table

S_id	S_Name	age
101	Adeel	15
102	Aamir	18
103	Amjad	17
104	Aslam	22

Like clause

- **Like** clause is used as condition in SQL query.
- Compares data with an expression using wildcard operators.
- Finding similar data from the table.
  - **Wildcard operators:**
  - **Percent sign** % : represents zero, one or more than one character.
  - **Underscore sign** _ : represents only one character.

Like clause

- Consider the following **Student** table.

s_id	s_Name	age
101	Adeel	15
102	Aamir	18
103	Amjad	17
104	Salman	18

- SELECT * from Student where s_name like 'A%';

s_id	s_Name	age
101	Adeel	15
102	Aamir	18
103	Amjad	17

Like clause

- SELECT * from Student where s_name like '_d%';
- The above query will return all records from **Student** table where **s_name** contain 'd' as second character.

s_id	s_Name	age
101	Adeel	15

- SELECT * from Student where s_name like '%r';The above query will return all records from **Student** table where **s_name** contain ‘r' as last character.

s_id	s_Name	age
102	Aamir	18

Order By Clause

- Order by clause is used with **Select** statement for arranging retrieved data in sorted order.
- by default sort data in ascending order.
- To sort data in descending order **DESC** keyword is used with **Order by** clause
  - *SELECT* column-list | * from table-name **order by** *asc|desc;*

Order By Clause

- Consider the following **Emp** table,

eid	name	age	salary
401	Amjad	22	9000
402	Shaheen	29	8000
403	Rehan	34	6000
404	Sameer	44	10000
405	Taimoor	35	8000

Order By Clause

- SELECT * from Emp **order by** salary;
- The above query will return result in ascending order of the **salary**.

eid	name	age	salary
403	Rehan	34	6000
402	Shaheen	29	8000
405	Taimoor	35	8000
401	Amjad	22	9000
404	Sameer	44	10000

Order By Clause

- □ Consider the **Emp** table described above,
- □ SELECT * from Emp order by salary DESC; The above query will return result in descending order of the **salary**.

eid	name	age	salary
404	Sameer	44	10000
401	Amjad	22	9000
405	Taimoor	35	8000
402	Shaheen	29	8000
403	Rehan	34	6000

HAVING Clause

- having clause is used with SQL Queries to give more precise condition for a statement.
- It is used to mention condition in Group based SQL functions, just like WHERE clause.
- select column_name, function(column_name) FROM table_name WHERE column_name condition GROUP BY column_name **HAVING** function(column_name) condition

HAVING Clause

- Consider the following **Sale** table.

oid	order_name	previous_balance	customer
11	ord1	2000	Aamir
12	ord2	1000	Adeel
13	ord3	2000	Amjad
14	ord4	2000	Adeel
15	ord5	2000	Aamir

- Suppose we want to find the customer whose previous_balance sum is more than 3000.
- SELECT * from sale group customer having sum(previous_balance) > 3000

oid	order_name	previous_balance	customer
11	ord1	4000	Aamir

Distinct keyword

- Retrieve unique values from the table removing all the duplicate records
- **SELECT** *distinct* column-name from *table-name;*
- Select distinct salary from emp;

eid	name	age	salary
401	Amjad	22	5000
402	Shaheen	29	8000
403	Rehan	34	10000
404	Sameer	44	10000
405	Taimoor	35	8000

salary
5000
8000
10000

AND & OR operator

- **AND** and **OR** operators are used with **Where** clause to make more precise conditions for fetching data from database by combining more than one condition together.

AND operator

- Consider the following **Emp** table

eid	name	age	salary
401	Amjad	22	5000
402	Shaheen	29	8000
403	Rehan	34	12000
404	Sameer	44	10000
405	Taimoor	35	9000

AND operator

- SELECT * from Emp WHERE salary < 10000 **AND** age > 25

eid	name	age	salary
402	Shaheen	29	8000
405	Taimoor	35	9000

OR operator

- Combining multiple conditions with *Where* clause
- The only difference between AND and OR is their behaviour.
- AND to combine two or more than two conditions, records satisfying all the condition
- OR, atleast one condition from the conditions specified must be satisfied by any record to be in the result.

OR operator

- Consider the following **Emp** table

eid	name	age	salary
401	Amjad	22	5000
402	Shaheen	29	8000
403	Rehan	34	12000
404	Sameer	44	10000
405	Taimoor	35	9000

OR operator

- □ SELECT * from Emp WHERE salary > 10000 **OR** age > 25 The above query will return records where either salary is greater than 10000 or age greater than 25.

402	Shaheen	29	8000
403	Rehan	34	12000
404	Sameer	44	10000
405	Taimoor	35	9000