

MD: Hasem Ali
University of Rajshahi
Roll: 2010777120

B.Sc. Engg. 4th Year Odd Semester 2015
DBMS LAB
DISCUSSION-1

Use the Pubs database and explore/perform the following topics/ tasks.

1. Knowing the names of all the tables in a database.

Discussion

The sysobjects system table holds data about all the tables and other database objects. You can write and execute the following command to see what this system table holds:

```
SELECT * FROM SYSOBJECTS
```

But only to get the table names, the more specific command is

```
SELECT NAME FROM SYSOBJECTS WHERE XTYPE='U'
```

2. Show all/specific records with all/ specific fields from a table.

Discussion

i) To show all the records with all the fields, the command is

```
SELECT * FROM <TABLENAME>
```

Example: To show all the records from the authors table.

```
SELECT * FROM AUTHORS
```

ii) To show all the records with specific fields, the command is,

```
SELECT <COLUMNNAME1, COLUMNNAME2.> FROM AUTHORS
```

Example: To show author last name and state all for authors from the authors table.

```
SELECT AU_LNAME, STATE FROM AUTHORS
```

iii) To show all fields of specific records satisfying a condition, the command is

```
SELECT * FROM AUTHORS WHERE <COLUMNNAME> <CONDITION>
```

Example: To show all the fields of those authors who live in the state of CA

```
SELECT * FROM AUTHORS WHERE STATE = 'CA'
```

Example: To show all the fields of those authors who have a last name 'White' and live in the state of CA

```
SELECT * FROM AUTHORS WHERE AU_LNAME='White' AND STATE = 'CA'
```

Use the Titles table for the following two tasks.

Task1: Show the name of the books which have yearly total sales of more than 8000.

Task2: Show the name of the books which have royalty of 12 to 24.

3. Showing ordered list

Discussion: Showing an ordered list. Titles tables sorted on the price field suppose we want to see the maximum price of the books. The command is

```
SELECT * FROM TITLES ORDER BY PRICE ASC
```

Also try DESC.

4. Showing aggregate values

Discussion: Suppose we want to see the maximum price of the books. The command is

SELECT MAX(PRICE) from TITLES

Similarly FOR AVERAGE price

SELECT AVG(PRICE) from TITLES

MAX(), AVG() are SQL functions.

5. Showing aggregate values within groups

Discussion: Suppose we want to show the book type and the average price of the each type. The command is

SELECT TYPE , MAX(PRICE) from TITLES GROUP BY TYPE

6. Showing aggregate values within groups having some condition

Discussion: Suppose we want to show the book type and the average price of the each type if average price is higher than a given values, say, 15. The command is

SELECT TYPE, AVG(PRICE) from TITLES GROUP BY TYPE HAVING AVG(PRICE) > 15.

Task3: • Show average price of the books of each type, the total yearly sales of that book type along with book type

7. Showing formatted string with customized header.

Discussion: We want to show the author name and ph no. where the name should be in the following manner,

J. White, i.e. First letter of first name dot last name

The command will be

SELECT "Name"=SUBSTRING(au_fname,1,1)) + '.' + au_lname, phone FROM authors

"Name" is the column heading on the output

Use the Pubs database and explore/perform the following topics/ tasks.

1. Joining Tables (Inner Join)

Discussion

The following query shows author's last name and title id of books.

SELECT au_lname, title_id FROM authors JOIN titleauthor ON authors.au_id=titleauthor.au_id

Task 1:

- i) Show the title of a book, the corresponding author name.
- ii) Show the title of a book, the corresponding author and publisher name.

2. The Cartesian product

The following query shows author's last name and city along with publisher's name and city

SELECT au_lname, pub_name FROM authors, publishers

Task 2:

- iii) Show the author name, city, publisher name and city only for which the authors and the publishers live in the same city

3. Nested query

SELECT * FROM titles WHERE royalty = (SELECT avg(royalty) FROM titles)

Task 3:

- iv) Show the author name(s) of the book which has the maximum royalty

Hint: use IN in place of =

4. Creating a table:

A table called CustomerAndSuppliers is to be created on the following schema

Field name	Data type	Size	Requirement
cust_id	Character	6	1. Primary key 2. Starting with C or S and then 5 digits i.e. C00001 or S0001.
cust_fname	character	15	NULL not allowed
cust_lname	character	15	Use variable character size
cust_address	text		
cust_telno	character	12	Must follow the format like 012-34567890
cust_city	character	12	Default value is Rajshahi
sales_amt	money		Negative values not allowed
proc_amt	money		Negative values not allowed

The corresponding SQL statement is as follows

CREATE TABLE CustomerAndSuppliers

```
(
    cust_id CHAR (6) PRIMARY KEY CHECK (cust_id LIKE '[CS][0-9][0-9][0-9][0-9][0-9]'),
    cust_name CHAR(15) NOT NULL,
    cust_name VARCHAR (15),
    cust_address TEXT,
    cust_telno CHAR (12) CHECK (cust_telno LIKE '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9]'),
    cust_city CHAR (12) DEFAULT 'Rajshahi',
    sales_amnt MONEY CHECK (sales_amnt >= 0),
    proc_amnt MONEY CHECK (proc_amnt >= 0)
)
```

5. Inserting data into a table:

Example:

INSERT CustomerAndSuppliers

(cust_id, cust_name, cust_name, cust_address, cust_telno, cust_city, sales_amnt, proc_amnt) VALUES ('C00001', 'Jibai', 'Hossain', '221/B Dhamondi', '017-00000000', 'Dhaka', 0, 0)

Task 4: Create tables on the following schema

Table name: Item

Field name	Data type	Size	Requirement
item_id	Character	6	1. Primary key 2. Starting with P and then 5 numbers i.e. A0001.
item_name	character	12	
item_category	character	10	Example: Electrical, Mechanical, Software, Books etc.
item_price	Float	12	Negative values not allowed
item_qoh	Integer		Negative values not allowed
item_last_sold	date		Default value is current date

Table: Transactions

Field name	Data type	Size	Requirement
tran_id	Character	10	1. Primary key 2. Starting with T and then 9 numbers i.e. T000000001.
item_id	character	6	Foreign key with reference to Item table
cust_id	character	6	Foreign key with reference to customer table
tran_type	character	1	Either S or O, (S for sales Order to supplier)
tran_quantity	Integer	12	Only positive values.
tran_date	date and time		Default is current date

DBMS LAB
DISCUSSION-3

1. ✓ Stored Procedure

Stored Procedures are database objects where multiple SQL statements can be executed as a batch. Stored procedures once created stays in the database and can be executed from client side.

The following stored procedure shows authors name of a given title_id

```
-CREATE PROC sp_showTitleAndAuthor
-AS
-BEGIN
-SELECT "Authors Last Name"=au_lname FROM authors where au_id in (select au_id from
-titleauthor where title_id='BU1032')
-END
```

To execute the just created Stored Procedure the command is

```
- EXEC sp_showTitleAndAuthor
```

To modify an existing stored procedure use the following statements

```
ALTER PROC sp_showTitleAndAuthor
AS
BEGIN
---
```

```
END
```

To delete the stored procedure from the database

```
DROP PROC sp_showTitleAndAuthor
```

2. ✓ Parameterized Stored procedure

Like function arguments Stored Procedures can accept values when being executed and can also return values.

Example: Modifying the procedure created in 1 that accepts an title_id and shows the corresponding author name

```
- ALTER PROC sp_showTitleAndAuthor @titleid char(15)
```

```
- AS
```

```
- BEGIN
```

```
- SELECT "Authors Last Name"=au_lname FROM authors where au_id in (select au_id from
```

- titleauthor where title_id=@titleid)

- END

EXEC sp_ShowTitleAndAuthor @titleid = 'BU1032'

3. Stored procedures with decision making/ looping constructs

The following procedure can be used to increase the price of a particular book by 10% but on the condition that the new price does not cross \$20

- CREATE PROC sp_updatePrice @titleid char(15)

- AS

- BEGIN

- DECLARE @price MONEY

-SELECT @price=price from TITLES WHERE title_id=@titleid

- set @price=@price+0.1*@price

-IF @price<=20

-UPDATE titles SET price= @price WHERE title_id=@titleid

- END

- EXEC sp_updatePrice 'BU7832'

Assignments

Using the tables created in the last class (i.e., CustomerAndSuppliers, Items, Transactions) perform the following tasks

✓ **Task 1:**

Write a stored procedure that prints out item categories, total number of items available and average price of that category in the following format.

<u>Category</u>	<u>Total number of Items</u>	<u>Average Price</u>
-----------------	------------------------------	----------------------

✓ **Task: 2**

Write a stored procedure that

a) accepts as two inputs, i.e., i) category name and ii) price value

b) And shows the item details under that category that are cheaper than the accepted price value,

✓ **Task 3:**

Write a stored procedure that

a) Accepts as input i) category name and ii) desired average price value

b) And increase the price of each item under that category by 10% until the new average price crosses the desired average price value.

1. Trigger

Trigger is special type of Stored Procedure which is attached to a table and is only executed (also known as fired) when an INSERT, UPDATE or DELETE (i.e., modification of table data) occurs on that table. One has to specify the modification action(s) that fire the trigger when it is created.

Unlike stored procedures, triggers can not be explicitly executed.

Example: 1 The following trigger shows a message when a row is inserted in Items table

```
CREATE TRIGGER trg_test ON Items FOR INSERT
```

```
AS
```

```
BEGIN
```

```
PRINT 'Data Inserted in Item Table'
```

```
END
```

2. Triggers can be used for automatically updating a table when an insert/update/delete statement takes place in another table. For example, whenever a transaction takes place, i.e., a row is inserted in the Transactions Table, item_qty field in the Items table should be updated either by increasing it or decreasing.

In such cases, a table called INSERTED is used by the trigger.

- INSERTED is a temporary table that SQL server automatically creates whenever an insertion takes place in a table. The inserted table holds only the row that is getting inserted.
- Similarly, a temporary table called DELETED consisting of row that is getting deleted is created when a deletion takes place.
- Both INSERTED and DELETED tables get created when an UPDATE statement takes place.

The following trigger updates Items table whenever a transaction takes place.

```
CREATE TRIGGER trg_update_item ON Transactions FOR INSERT
```

```
AS
```

```
BEGIN
```

```
DECLARE @item_id char(6), @tranamount int, @tran_type char(1)
```

```
SELECT @item_id=item_id, @tranamount=tran_quantity, @tran_type=tran_type FROM INSERTED
```

```
IF (@tran_type = 'S')
```

```
UPDATE Items SET item_qty=item_qty- @tranamount WHERE item_id=@item_id
```

```
ELSE
```

```
UPDATE Items SET item_qty=item_qty+ @tranamount WHERE item_id=@item_id
```

```
END
```

Assignment

Task 1: Write a trigger on Transaction that automatically updates the sold_amnt or proc_amnt field of CustomersAndSuppliers table whenever a transaction happens.