Database Systems Laboratory 5

Grouping, DCL & TCL

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

Grouping, DCL & TCL

1. Grouping Data
2. Data Control Language (DCL)
3. Transaction Control Language (TCL)
4. SQL*PLUS Commands

Grouping Data

GROUP BY clause

GROUP BY clause is used for grouping data. The column appearing in SELECT statement must also appear in GROUP BY clause. The syntax of this is:
SELECT column, groupfunction(column) FROM tablename GROUP BY column;

SELECT DEPT_NO FROM EMP GROUP BY DEPT_NO;

DEPT_NO
30
20
10

SELECT DEPT_NO, AVG(SAL) FROM EMP GROUP BY DEPT_NO;

DEPT_NO	AVG(SAL)
30	1566.6666666666666666666666666666666667
20	2175
10	2926.6666666666666666666666666666666667

Grouping Data...

HAVING Clause

HAVING clause can restrict groups. The syntax of this is:

SELECT column, groupfunction(column) FROM tablename GROUP BY column [HAVING groupcondition];

SELECT DEPT_NO, AVG(SAL) FROM EMP GROUP BY DEPT_NO HAVING AVG(SAL)>2000;

DEPT_NO	AVG(SAL)
20	2175
10	2926.6666666666666666666666666666666667

Grouping Data...

WHERE and GROUP BY Clause

The data can be restricted before the grouping by using WHERE clause. The syntax of this is:

SELECT column, groupfunction(column) FROM tablename [WHERE condition] GROUP BY column;

SELECT DEPT_NO, AVG(SAL) FROM EMP WHERE SAL>2000 GROUP BY DEPT_NO;

DEPT_NO	AVG(SAL)
30	2850
20	2991.6666666666666666666666666666666667
10	3740

Grouping Data...

WHERE, GROUP BY and HAVING Clause

WHERE clause is used to filter the unwanted records, while the HAVING clause is used to filter the unwanted groups. The syntax for it is:

SELECT column, groupfunction(column) FROM tablename [WHERE condition] GROUP BY column [HAVING groupcondition];

SELECT DEPT_NO, AVG(SAL) FROM EMP WHERE SAL>2000 GROUP BY DEPT_NO HAVING AVG(SAL)<3000;

DEPT_NO	AVG(SAL)
30	2850
20	2991.666666666666666666666666666666667

Data Control Language

An object privilege specifies what a user can do with a database object, such as a table or a view. The different privileges for table are: ALTER, INSERT, UPDATE, DELETE, and SELECT

Granting Privileges

A user can grant privileges on objects from own schema to other users or roles by using GRANT command. The syntax of providing a privilege is:

GRANT privileges/ ALL ON objectname TO username/ PUBLIC [WITH GRANT OPTION];

WITH GRANT OPTION clause allows the grantee to grant privileges to other users and roles

GRANT SELECT, INSERT ON Employee TO Mita;

GRANT SELECT, INSERT ON Employee TO Mita WITH GRANT OPTION;

Data Control Language...

*SELECT * FROM SYSTEM.Employee;*

GRANT SELECT ON SYSTEM.Employee TO Mithun;

Revoking Privileges

If a user granted privileges by a WITH GRANT OPTION to another user and that second user passed on those privileges, the REVOKE statement takes privileges not only from the grantee but also from the users granted privileges by the grantee. The general syntax is:
REVOKE privileges/ALL ON objectname FROM username/ PUBLIC;

REVOKE INSERT ON Employee FROM Mita;

REVOKE ALL ON Employee FROM Mita;

Transaction Control Language

A transaction consists of a sequence of query and update statements. Transaction Control Language gives you flexibility to undo transactions or write transactions to the disk. Transactions provide consistency in case of a system failure

Committing a transaction

COMMIT statement is used to end the current transaction and makes permanent any changes made during transaction. The general syntax is:

COMMIT;

Roll backing the operations

ROLLBACK statement is used to discard parts or all of the work the user has done in the current transaction. The syntax for this is:

ROLLBACK;

Transaction Control Language...

Roll backing the operations to a particular position

SAVEPOINT allows the user to create logical marking in the whole transaction, so that the system will discard all the changes up to a point. The syntax for creating a SAVEPOINT is:
SAVEPOINT savepointname;

The syntax for roll backing to a particular savepoint is:
ROLLBACK TO SAVEPOINT savepointname;

SQL*PLUS Commands

SAVE filename.sql [REPLACE/ APPEND]

It saves current buffer contents to a file

GET filename.sql

It retrieves previously saved file to the buffer

START filename.sql

It runs a previously saved command from file

@filename.sql

Same as START command

The filename in the file-related commands requires entire file path

SQL*PLUS Commands...

EDIT

It opens the current buffer named *afiedt.buf*

EDIT filename.sql

It allows to edit a saved file

PASSWORD

It allows to change the password

CREATE USER username IDENTIFIED BY password

It creates a new user with respective password

EXIT

It leaves SQL *PLUS environment & commits current transaction