Lab 07

Data manipulation operations in SQL

SQL Developer

We are giving a small touch of SQL developer in this class to create DDL scripts for our scott schema tables as we are going to Modify them

1

SQL Developer

SQL Developer

In class we have seen demo of SQL Developer basics, how to create DDL scripts and data scripts by using SQL Developer (any tool can be used)

1

Data Manipulation Language

Data Manipulation Language

When we want to add, update or delete data in the database, we execute a DML statement.

Transaction

A collection of DML statements that form a logical unit of work.

If you are a bank customer having 2 accounts at a time let say in account 1 or account 2 if you deposit money in account 2 from account 1 then account 1 will show decrease amount transaction ,account 2 will show increase amount transaction and there might be a transaction log for a user which will show you log of both transactions.

Data Manipulation Language

SQL DML Statements

Notation	Description
INSERT	Enter new rows into tables
UPDATE	To change existing rows
DELETE	To delete existing rows

2

Adding a new Row to a Table

Adding a Row to a Table

Adding a new Row to a table

INSERT Statement is used for adding a new row into a table

Syntax

INSERT INTO table [(column [, column ...])]
VALUES (value [, value ...]);

Example 1: Insert a new Row in department table as management decide to start a new department in ***Detroit(location)*** named as ***DEVELOPMENT dept. # allocated to that department is 50.***

```
INSERT INTO dept (deptno, dname, loc)
VALUES (50, 'DEVELOPMENT', 'DETROIT');
```

Adding a Row to a Table

```
SQL> ed
Wrote file afiedt.buf

  1  INSERT INTO dept (deptno, dname, loc)
  2* VALUES (50, 'DEVELOPMENT', 'DETROIT')
SQL> /

1 row created.
```

```
SQL> select * from dept
  2  ;

    DEPTNO DNAME          LOC
__________ _______________ _____________
        50 DEVELOPMENT    DETROIT
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

Newly entered Record

Note If the column list is not mentioned than all columns must be listed in the default order as they are present in the table(see via desc table command)

```
SQL> desc dept;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ---------------------
 DEPTNO                                                NOT NULL NUMBER(2)
 DNAME                                                          VARCHAR2(14)
 LOC                                                            VARCHAR2(13)
```

So insert statement can be written as:

INSERT INTO dept VALUES (50, ‘DEVELOPMENT’, ‘DETROIT’);

Adding a Row to a Table

Inserting rows with Null values

Either we explicitly Enter ***all columns according to table order*** or ***with column names*** via mentioning **Null** values for respective Null Columns this is called Explicit Method

```
INSERT INTO dept VALUES (70, 'FINANCE', NULL);
```

OR

```
INSERT INTO dept (deptno, dname, loc)
VALUES (70, 'FINANCE', NULL);
```

Or we Implicitly Omit the columns (to whom we want to set Null) after only Entering ***all columns with column names*** this is called Implicit Method

```
INSERT INTO dept (deptno, dname)
VALUES (60, 'MIS');
```

Location will set to Null

Adding a Row to a Table

Enforcement of All data types ,data ranges and integrity constraints is automatically applied in Oracle so users cant violate them on insertion

Example 2: today company hired an employee Mr ***Green*** as ***SALESMAN*** with empno ***7196*** in Accounting department (deptno ***10***) whose Manager is Mr. Clark (empid is ***7782***).Company decided to give him salary **2000** per month with ***no commission***.

INSERT INTO emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)
VALUES (7196, 'GREEN', 'SALESMAN', 7782, SYSDATE, 2000, NULL, 10)

Adding a Row to a Table

Date function can also been used for inserting specific date value

Example 3: On **03rd Feb 1997** company hired an employee Mr ***Shaam*** *as* ***Analyst*** with empno ***2296*** in Accounting department (deptno ***10***) whose Manager is Mr. Clark (empid is ***7782***).Company decided to give him salary **3000** per month with ***no commission***.

```
INSERT INTO emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)
VALUES (2296, 'SHAAM', 'ANALYST', 7782, TO_DATE('FEB 3, 97',
'MON DD, YY'), 3000, NULL, 10)
```

Let see all records of the Employee table

Adding a Row to a Table

```
SQL> select * from emp;

     EMPNO ENAME      JOB              MGR HIREDATE          SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7196 GREEN      SALESMAN        7782 27-AUG-16       2000                    10
      2296 SHAAM      ANALYST         7782 03-FEB-97       3000                    10
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

Examples 2
Examples 3

Hiredate saved properly.

Adding a new Row Using substitution variable

```
INSERT INTO dept (deptno, dname, loc)
VALUES (&department_id, '&department_name', '&location');
```

Adding a Row to a Table

```
  1  INSERT INTO dept (deptno, dname, loc)
  2* VALUES (&department_id, '&department_name', '&location')
SQL> /
Enter value for department_id: 80
Enter value for department_name: EDUCATION
Enter value for location: ATLANTA
old   2: VALUES (&department_id, '&department_name', '&location')
new   2: VALUES (80, 'EDUCATION', 'ATLANTA')

1 row created.
```

Copy one row of one table to another (same table structures)

To implement this scenario lets create a **LoyalEmp** table as we have already created DDL of emp table lets create this new table using that DDL script and assume that a list of loyal employees are maintained by a company in a that new table.

3

Changing Data in a table (Update)

Changing data in a table (update)

Update Statement is used for changing data in a table

Syntax

UPDATE table
SET column = value [, column = value , …]
[WHERE condition];

This shows that multiple column values can be updated at a same time

This shows that multiple Rows can Be updated depending on the condition

Example 3: an employee Mr. Clarke recently shifted to a Research department (dept no =20) from accounting. His record needs to be updated.

```
UPDATE emp
SET deptno = 20
WHERE empno = 7782
```

Changing data in a table (update)

If no where clause is mention in a update query then it will update all rows of the table. If I run the above query in **Loyalemp** table

Before Update

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7196	GREEN	SALESMAN	7782	27-AUG-16	2000		10
2296	SHAAM	ANALYST	7782	03-FEB-97	3000		10
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7839	KING	PRESIDENT		17-NOV-81	5000		10
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

```
UPDATE Loyalemp
SET deptno = 20;
```

After Update

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7196	GREEN	SALESMAN	7782	27-AUG-16	2000		20
2296	SHAAM	ANALYST	7782	03-FEB-97	3000		20
7782	CLARK	MANAGER	7839	09-JUN-81	2450		20
7839	KING	PRESIDENT		17-NOV-81	5000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		20

Changing data in a table (update)

Updating with Multiple column subquery

Example 3: lets experiment in our **Loyalemp** table that Mr Miller(7934) ***JOB,Deptno & MGR*** needs to be replace with the Emp's table Mr Blakes(7698) ***JOB,Deptno & MGR***

```
UPDATE loyalemp
SET (job, deptno,MGR) =
                (SELECT job, deptno,MGR
                FROM emp
                WHERE empno = 7698)
WHERE empno = 7934;
```

```
SQL> select * from loyalemp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7196 GREEN      SALESMAN        7782 27-AUG-16       2000                    20
      2296 SHAAM      ANALYST         7782 03-FEB-97       3000                    20
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    20
      7934 MILLER     MANAGER         7839 23-JAN-82       1300                    30
```

Updated Cells

4

Removing a Row from Table (Delete)

Removing Row from Table (Delete)

Delete Statement is used for Removing existing row from a table

Syntax

Delete From table
[WHERE condition];

Example 1: Remove the **development** department from Departments table

DELETE FROM dept
WHERE dname = 'DEVELOPMENT';

```
select * from dept;

DEPTNO DNAME          LOC
------ -------------- ----------
    80 EDUCATION      ATLANTA
    10 ACCOUNTING     NEW YORK
    20 RESEARCH       DALLAS
    30 SALES          CHICAGO
    40 OPERATIONS     BOSTON
```

Removing Row from Table (Delete)

All rows of the table will be removed if we will omit where clause

Example 2: Remove all rows from loyalemp table

DELETE FROM loyalemp ;

```
SQL> select * from loyalemp;

no rows selected
```

Example 3: copy all employees into loyal employees table then remove all those employees who were hired after January 1,1977.

Step 1:

INSERT INTO loyalemp

Select * from emp

Step 2:

DELETE FROM loyalemp

WHERE hiredate > TO_DATE('01.01.97', 'DD.MM.YY');

Removing Row from Table (Delete)

Before Deletion

```
SQL> select * from loyalemp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7196 GREEN      SALESMAN        7782 22-AUG-16       2000                    10
      2296 SHAAM      ANALYST         7782 03-FEB-97       3000                    10
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

16 rows selected.
```

Mr. shaam record has been removed

After Deletion

```
SQL> select * from loyalemp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7196 GREEN      SALESMAN        7782 27-AUG-16       2000                    10
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

15 rows selected.
```

Delete Rows via subquery

Example 3: Delete all loyal employees of sales department

```
DELETE from loyalemp
WHERE deptno =
      (SELECT deptno
      FROM dept
      WHERE dname = 'SALES');
```

```
  1   DELETE from loyalemp
  2   WHERE deptno =
  3   (SELECT deptno
  4   FROM dept
  5*  WHERE dname = 'SALES')
  6   /

6 rows deleted.
```

```
     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7196 GREEN      SALESMAN        7782 27-AUG-16       2000                    10
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

9 rows selected.
```

4

Database Transactions

Database Transaction

Concept of Database transaction is to maintain consistency that is based on multiple transactions at a time with in a single operation. We know that transaction consist of DML statements that will finally impact one consistent change to the whole data. To maintain that consistency we must be sure that either all of the transactions successfully done(commit) or if few are done or few are not due to some technical issues then transaction must be Rollback to run the whole process again.

Example : funds transfer between two accounts includes debit from one account and credit to another account. Although they are 2 different queries but they both **either run successfully** or if due to some issue debit query successfully run but credit query was unable to run then either of the query will not impact to the tables to maintain consistency *unless commit command is executed.* This scenario will be *Rollback.*

Database Transaction

Transaction Types

Notation	Description
Data Manipulation Language(DML)	Consists of any number of DML statements that the Oracle Server treats as a single entity or a logical unit of work after commit.
Data Definition Language(DDL)	Consists of only one DDL statement
Data Control Language(DCL)	Consists of only one DCL statement

DCL (data Control language) is used to control privileges in Database. Let say to allow a user to create table we write following DCL command:

grant *create table* to username;

Similarly to take create table rights we write:

Revoke *create table* from username;

Database Transaction

A transaction begin when the first SQL statement is encountered and then transaction terminates when:

1. ***commit*** or ***Rollback*** statement is issued
2. User exits ***SQL Plus*** or ***any*** database session
3. Machine fails or a system crashes

Otherwise as soon as one transaction ends the next statement automatically starts the next transaction.

As far as DDL or DCL statements are concern they are single statements and are committed automatically with end of transaction

Transaction Control(TCL Commands)

COMMIT: Ends the current transaction by making all pending data changes permanent

ROLLBACK: Ends the current transaction by discarding all pending data changes

Database Transaction

Transaction Control (cont.)

Savepoint: Instead of doing complete rollback at any abnormal terminating event we can be partially rollback by setting savepoints in our code.

Database Transaction

Example 3: create a new department 'Advertising' with 'Atlanta' as a location with deptno = 50 than assign this department to Mr jones of LoyalEmp table.

```
INSERT INTO dept (deptno, dname, loc)
VALUES (50, 'ADVERTISING', 'ATLANTA');
```

Step 1

```
UPDATE LOYALEMP
SET DEPTNO = 50
WHERE EMPNO = 7566;
```

Step 2

```
COMMIT;
```

Step 3